REPÚBLICA DEL PERÚ
PERÚ
Ministerio
de Educación

Book of Syllabi

School of Computer Science

– 2021-I–

Lima: July 7, 2020

Task Force

Ernesto Cuadros-Vargas (Editor) <*ecuadros@spc.org.pe*>
President of the Peruvian Computer Society (SPC) 2001-2007, 2009
Member of the Steering Committee de ACM/IEEE-CS Computing Curricula
for Computer Science (CS2013)
Member of Steering Committee de ACM/IEEE-CS Computing Curricula 2020
(CS2020)
http://socios.spc.org.pe/ecuadros

Contents

First Semester **5**

Second Semester **31**

Third Semester **55**

Fourth Semester **86**

Fifth Semester **110**

Sixth Semester **148**

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS111. Introduction to Computer Science (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the first course in the sequence of introductory courses to Computer Science.This course is intended to cover the concepts outlined by the Computing Curricula IEEE-CS/ACM 2013. Programming is one of the pillars of Computer Science; any professional of the area, will need to program to materialize their models and proposals. This course introduces participants to the fundamental concepts of this art. Topics include data types, control structures, functions, lists, recursion, and the mechanics of execution, testing, and debugging.

5. GOALS

- Introduce the fundamental concepts of programming.
- Develop the ability of abstraction using programming language

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

a10) (**10**)

a11) (**11**)

b1) (**1**)

d1) (**1**)

7. TOPICS

<table>
<tr><td colspan="2"><b>Unit 1: History (5)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: a</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• Prehistory, the world before 1946
• History of computer hardware, software, networking
• Pioneers of computing
• History of the Internet</td><td>• Identify significant continuing trends in the history of the computing field [Familiarity]
• Identify the contributions of several pioneers in the computing field [Familiarity]
• Discuss the historical context for several programming language paradigms [Familiarity]
• Compare daily life before and after the advent of personal computers and the Internet [Assessment]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [BB19], [Gut13], [Zel10]</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Unit 2: Basic Type Systems (2)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: a</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• A type as a set of values together with a set of operations
– Primitive types (e.g., numbers, Booleans)
– Compound types built from other types (e.g., records, unions, arrays, lists, functions, references)
• Association of types to variables, arguments, results, and fields
• Type safety and errors caused by using values inconsistently given their intended types</td><td>• For both a primitive and a compound type, informally describe the values that have that type [Familiarity]
• For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Familiarity]
• Describe examples of program errors detected by a type system [Familiarity]
• For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage]
• Use types and type-error messages to write and debug programs [Usage]
• Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Gut13], [Zel10]</td></tr>
</table>

Unit 3: Fundamental Programming Concepts (9)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing • The concept of recursion	• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Assessment] • Identify and describe uses of primitive data types [Familiarity] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Familiarity] • Describe the concept of recursion and give examples of its use [Assessment] • Identify the base case and the general case of a recursively-defined problem [Familiarity]
Readings : [Gut13], [Zel10]	

Unit 4: Basic Analysis (2)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm • Big O notation: formal definition • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Big O notation: use • Analysis of iterative and recursive algorithms	• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Familiarity] • In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors [Familiarity] • State the formal definition of big O [Familiarity] • Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms [Usage] • Use big O notation formally to give expected case bounds on time complexity of algorithms [Usage]
Readings : [Gut13], [Zel10]	

Unit 5: Fundamental Data Structures and Algorithms (8)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort) • Hash tables, including strategies for avoiding and resolving collisions • Binary search trees – Common operations on binary search trees such as select min, max, insert, delete, iterate over tree • Graphs and graph algorithms – Representations of graphs (e.g., adjacency list, adjacency matrix) – Depth- and breadth-first traversals • Heaps • Graphs and graph algorithms – Maximum and minimum cut problem – Local search • Pattern matching and string/text algorithms (e.g., substring matching, regular expression matching, longest common subsequence algorithms)	• Implement basic numerical algorithms [Usage] • Implement simple search algorithms and explain the differences in their time complexities [Assessment] • Be able to implement common quadratic and O(N log N) sorting algorithms [Usage] • Describe the implementation of hash tables, including collision avoidance and resolution [Familiarity] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Familiarity] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity] • Explain how tree balance affects the efficiency of various binary search tree operations [Familiarity] • Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Usage] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment] • Describe the heap property and the use of heaps as an implementation of priority queues [Familiarity] • Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Usage] • Trace and/or implement a string-matching algorithm [Usage]
Readings : [Gut13], [Zel10]	

Unit 6: Algorithms and Design (9)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• The concept and properties of algorithms – Informal comparison of algorithm efficiency (e.g., operation counts) • The role of algorithms in the problem-solving process • Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • Fundamental design concepts and principles – Abstraction – Program decomposition – Encapsulation and information hiding – Separation of behaivor and implementation	• Discuss the importance of algorithms in the problem-solving process [Familiarity] • Discuss how a problem may be solved by multiple algorithms, each with different properties [Familiarity] • Create algorithms for solving simple problems [Usage] • Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage] • Implement, test, and debug simple recursive functions and procedures [Usage] • Determine whether a recursive or iterative solution is most appropriate for a problem [Assessment] • Implement a divide-and-conquer algorithm for solving a problem [Usage] • Apply the techniques of decomposition to break a program into smaller pieces [Usage] • Identify the data components and behaviors of multiple abstract data types [Usage] • Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage] • Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Assessment]
Readings : [Gut13], [Zel10]	

Unit 7: Development Methods (1)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Modern programming enviroments – Code search – Programming using library components and their APIs	• Construct and debug programs using the standard libraries available with a chosen programming language [Familiarity]
Readings : [Gut13], [Zel10]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[BB19] J. Glenn Brookshear and Dennis Brylow. *Computer Science: An Overview.* Ed. by PEARSON. Global Edition. Pearson, 2019. ISBN: 1292263423. URL: http://www.pearsonhighered.com/brookshear.

[Gut13] John V Guttag. . *Introduction To Computation And Programming Using Python.* MIT Press, 2013.

[Zel10] John Zelle. *Python Programming: An Introduction to Computer Science.* Franklin, Beedle & Associates Inc, 2010.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS1D1. Discrete Structures I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Discrete structures provide the theoretical foundations necessary for computation. These fundamentals are not only useful to develop computation from a theoretical point of view as it happens in the course of computational theory, but also is useful for the practice of computing; In particular in applications such as verification, cryptography, formal methods, etc.

5. GOALS

- Apply Properly concepts of finite mathematics (sets, relations, functions) to represent data of real problems.
- Model real situations described in natural language, using propositional logic and predicate logic.
- Determine the abstract properties of binary relations.
- Choose the most appropriate demonstration method to determine the veracity of a proposal and construct correct mathematical arguments.
- Interpret mathematical solutions to a problem and determine their reliability, advantages and disadvantages.
- Express the operation of a simple electronic circuit using Boolean algebra.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a1) (**1**)

a2) (**2**)

a3) (**3**)

j1) (**1**)

j2) (**2**)

7. TOPICS

Unit 1: Sets, Relations, and Functions (22)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Sets – Venn diagrams – Union, intersection, complement – Cartesian product – Power sets – Cardinality of finite sets • Relations: – Reflexivity, simmetry, transitivity – Equivalence relations – Partial order relations and sets – Extremal elements of a partially ordered sets • Functions – Surjections, injections, bijections – Inverses – Composition	• Explain with examples the basic terminology of functions, relations, and sets [Assessment] • Perform the operations associated with sets, functions, and relations [Assessment] • Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context [Assessment]
Readings : [Gri03], [Ros07], [Vel06]	

Unit 2: Basic Logic (14)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Propositional logic • Logical connectives • Truth tables • Normal forms (conjunctive and disjunctive) • Validity of well-formed formula • Propositional inference rules (concepts of modus ponens and modus tollens) • Predicate logic – Universal and existential quantification • Limitations of propositional and predicate logic (e.g., expressiveness issues)	• Convert logical statements from informal language to propositional and predicate logic expressions [Usage] • Apply formal methods of symbolic propositional and predicate logic, such as calculating validity of formulae and computing normal forms [Usage] • Use the rules of inference to construct proofs in propositional and predicate logic [Usage] • Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis (eg, program correctness), database queries, and algorithms [Familiarity] • Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles [Usage] • Describe the strengths and limitations of propositional and predicate logic [Usage]
Readings : [Ros07], [Gri03], [Vel06]	

Unit 3: Proof Techniques (14)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Notions of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction • The structure of mathematical proofs • Direct proofs • Disproving by counterexample • Proof by contradiction • Induction over natural numbers • Structural induction • Weak and strong induction (i.e., First and Second Principle of Induction) • Recursive mathematical definitions • Well orderings	• Identify the proof technique used in a given proof [Assessment] • Outline the basic structure of each proof technique (direct proof, proof by contradiction, and induction) described in this unit [Usage] • Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument [Usage] • Determine which type of proof is best for a given problem [Assessment] • Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures [Familiarity] • Explain the relationship between weak and strong induction and give examples of the appropriate use of each [Assessment] • State the well-ordering principle and its relationship to mathematical induction [Familiarity]
Readings : [Ros07], [**Vel06**], [Sch12], [Vel06]	

Unit 4: Data Representation (10)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Numerical representation: sign-magnitude, floating point. • Representation of other objects: sets, relations, functions.	• Explain numerical representations such as sign-magnitude and floating point. [Assessment]. • Carry out arithmetic operations using different kinds of representations. [Assessment]. • Explain the floating point standard IEEE-754 [Familiarity].
Readings : [Ros07], [Gri03], [Vel06]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction.* 5 ed. Pearson, 2003.

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications.* 7 ed. Mc Graw Hill, 2007.

[Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction.* 3 ed. Brooks Cole, 2012.

[Vel06] Daniel J. Velleman. *How to Prove It: A Structured Approach.* Ed. by Cambridge University Pres. 2nd. 2006. ISBN: 978-0521675994.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

MA100. Mathematics I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	5
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The course aims to develop in students the skills to deal with models in science and engineering related to single variable differential calculus skills. In the course it is studied and applied concepts related to calculation limits, derivatives and integrals of real and vector functions of single real variables to be used as base and support for the study of new contents and subjects. Also seeks to achieve reasoning capabilities and applicability to interact with real-world problems by providing a mathematical basis for further professional development activities.

5. GOALS

- .
- .
- .

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: (20)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• . • .	• . • .
Readings : [Ste12], [ión14]	

Unit 2: (10)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• . • . • . • . • . • .	• . • . • . • . • . • .
Readings : [Ste12], [ión14]	

Unit 3: (20)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• . • . • . • . • .	• . • . • . • . • . • . • . • . • . • . • .
Readings : [Ste12], [ión14]	

Unit 4: (22)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• . • . • . • .	• . • . • . • . • . • . • . • . • . • . • .
Readings : [Ste12], [ión14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[ión14] ROn Larson ión. *Calculus*. 10th. 2014.

[Ste12] James Stewart. *Calculus*. 7th. 2012.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

FG101. Communication (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Para lograr una eficaz comunicación en el ámbito personal y profesional, es prioritario el manejo adecuado de la Lengua en forma oral y escrita. Se justifica, por lo tanto, que los alumnos de la Universidad Católica San Pablo conozcan, comprendan y apliquen los aspectos conceptuales y operativos de su idioma, para el desarrollo de sus habilidades comunicativas fundamentales: Escuchar, hablar, leer y escribir. En consecuencia el ejercicio permanente y el aporte de los fundamentos contribuyen grandemente en la formación académica y, en el futuro, en el desempeño de su profesión

In order to achieve effective communication in the personal and professional field, the proper handling of the Language in oral and written form is a priority. It is therefore justified that the students of UTEC University know, understand and apply the conceptual and operational aspects of their language, for the development of their fundamental communicative skills: Listening, speaking, reading and writing. Consequently the permanent exercise and the contribution of the fundamentals contribute greatly in the academic formation and, in the future, in the performance of his profession.

5. GOALS

- Desarrollar capacidades comunicativas a través de la teoría y práctica del lenguaje que ayuden al estudiante a superar las exigencias académicas del pregrado y contribuyan a su formación humanística y como persona humana.

- Develop communicative skills through the theory and practice of language that help the student to overcome the academic requirements of the undergraduate and contribute to his humanistic training and human person.

6. COMPETENCES

n) Apply knowledge of the humanities in their professional work. (**Usage**)

ñ) Understand that the formation of a good professional is not disconnected or opposed but rather contributes to genuine personal growth. This requires the assimilation of solid values, broad spiritual horizons and a deep vision of the cultural environment. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: (16)	
Competences Expected: C17,C20	
Topics	**Learning Outcomes**
• La comunicación, definición, relevancia. Elementos. Proceso. Funciones. Clasificación.Comunicación oral y escrita. • El lenguaje: definición. Características y funciones. Lengua: niveles. Sistema. Norma. Habla. El signo lingüístico: definición, características. • Multilingüismo en el Perú. Variaciones dialectales en el Perú. • La palabra: definición, clases y estructura. Los monemas: lexema y morfema. El morfema: clases. La etimología. • El Artículo académico: Definición, estructura, elección del tema, delimitación del tema. • The communication, definition, relevance. Elements. Process. Functions. Classification. Oral and written communication. • The language: definition. Features and functions. Language: levels. System. Rule. Speaks. The linguistic sign: definition, characteristics. • Multilingualism in Peru. Dialect variations in Peru. • The word: definition, classes and structure. The monemas: lexema and morpheme. The morpheme: classes. Etymology. • The Academic Article: Definition, structure, choice of topic, delimitation of the topic.	• Reconocer y valorar la comunicación como un proceso de comprensión e intercambio de mensajes, diferenciando sus elementos, funciones y clasificación [Usage]. • Analizar las características, funciones y elementos del lenguaje y de la lengua [Usage]. • Identificar las características del multilingüismo en el Perú, valorando su riqueza idiomática [Usage]. • Identificar las cualidades de la palabra y sus clases [Usage]. • Recognize and value communication as a process of understanding and exchanging messages, differentiating its elements, functions and classification[Usage]. • Analyze the characteristics, functions and elements of language and language [Usage]. • Identify the characteristics of multilingualism in Peru, valuing its idiomatic richness [Usage]. • Identify the qualities of the word and its classes [Usage].
Readings : [Len10]	

Unit 2: (16)	
Competences Expected: C17, C24	
Topics	**Learning Outcomes**
• Párrafo: Idea principal, secundaria y global. • El texto: definición, características. Cohesión y coherencia. • Organización del texto: La referencia (deixis); anáfora, catáfora, elipsis. Conectores lógicos y textuales. • Tipos de texto: descriptivo (procesos), expositivo, argumentativo. • Funciones de elocución en el texto: generalización, identificación, nominalización, clasificación, ejemplificación, definición. • Textos discontinuos: gráficos, tablas y diagramas. • Búsqueda de información. Fuentes de información. Referencias y citas. Registro de información: fichas, notas, resúmenes, etc. Aparato crítico: concepto y finalidad. Normas APA u otro. • Paragraph: Main, secondary and global idea. • The text: definition, characteristics. Cohesion and coherence. • Organization of the text: The reference (dejis); Anaphora, cataphora, ellipsis. Logical and textual connectors. • Types of text: descriptive (processes), expository, argumentative. • Functions of elocution in the text: generalization, identification, nominalization, classification, exemplification, definition. • Discontinuous texts: graphs, tables and diagrams. • Search for information. Information sources. References and citations. Record of information: index cards, notes, summaries, etc. Critical apparatus: concept and purpose. APA Standards or other.	• Redactar textos expositivos resaltando la idea principal y secundaria [Usage]. • Redactar textos expositivos con adecuada cohesión y coherencia, haciendo uso de referencias y conectores textuales [Usage]. • Interpretar textos discontinuos valorando su importancia para la comprensión del mensaje [Usage]. • Redactar textos expositivos resaltando la idea principal y secundaria [Usage]. • Redactar textos expositivos con adecuada cohesión y coherencia, haciendo uso de referencias y conectores textuales [Usage]. • Interpretar textos discontinuos valorando su importancia para la comprensión del mensaje [Usage].
Readings : [Len10], [Gat07]	

Unit 3: (12)	
Competences Expected: C17	
Topics	**Learning Outcomes**
• La oración: definición y clases. La oración enunciativa, interrogativa, imperativa, exclamativa, optativa. La proposición y la frase. La oración simple y compuesta. Coordinación y subordinación. El sintagma: estructura y clases: nominal, verbal, adjetival, preposicional, adverbial. • Elaboración de un glosario de términos técnicos, abreviaturas y siglas relacionadas con la especialidad (actividad permanente a lo largo del semestre). • Redacción del artículo académico: Resumen, palabras clave, introducción, desarrollo, conclusiones, bibliografiaTecnologia (Normas APA u otro que la Escuela profesional requiera).	• Reconocer y analizar la estructura oracional valorando su importancia y utilidad en la redacción de textos [Usage]. • Registrar y emplear terminología propia de la especialidad [Usage].
Readings : [San05]	

Unit 4: (12)	
Competences Expected: C17, C20, C24	
Topics	**Learning Outcomes**
• Redacción de correspondencia: carta - solicitud, informe, memorando, hoja de vida. • El discurso oral: propósitos, partes. Escuchar: propósitos y condiciones. Vicios de dicción: barbarismo, solecismo, cacofonía, redundancia, anfibología, monotonía. Régimen preposicional. • Comunicación en grupo Proceso, dinámica, estructura Formas (Técnicas): Mesa redonda, panel, foro y debate. • Revisión final del artículo académico. Presentación y exposición oral de trabajos de producción intelectual.	• Redactar textos académicos y funcionales atendiendo los distintos momentos de su producción, su estructura, finalidad y formalidad [Usage]. • Demostrar habilidades como emisor o receptor en distintas situaciones de comunicación con corrección idiomática [Usage]. • Aplicar las diferentes formas (técnicas) de comunicación en grupo reconociendo su importancia para la solución de problemas, toma de decisiones o discusión [Usage].
Readings : [Mar06]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Gat07] Carlos Gatti Muriel. *Elementos de la gramática española*. Lima, Universidad del Pacífico., 2007.

[Len10] Real Academia de la Lengua Española. *Nueva gramática de la lengua española, morfología y sintaxis*. Madrid, España: Ed. Espasa, 2010.

[Mar06] Gonzalo Martin Vivaldi. *Teoría y práctica de la composición y estilo*. Thompson, 2006.

[San05] J Sanchez Lobato. *Saber Escribir*. España, Instituto Cervantes, 2005.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

FG102. Study Methodology (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Los alumnos en formación profesional necesitan mejorar su actitud frente al trabajo y exigencia académicos. Además conviene que entiendan el proceso mental que se da en el ejercicio del estudio para lograr el aprendizaje; así sabrán dónde y cómo hacer los ajustes más convenientes a sus necesidades. Asimismo, requieren dominar variadas formas de estudiar, para que puedan seleccionar las estrategias más convenientes a su personal estilo de aprender y a la naturaleza de cada asignatura. De igual modo conocer y usar maneras de buscar información académica y realizar trabajos creativos de tipo académico formal, así podrán aplicarlos a su trabajo universitario, haciendo exitoso su esfuerzo.

5. GOALS

- Desarrollar en el estudiante actitudes y habilidades que promuevan la autonomía en el aprendizaje, el buen desempeño académico y su formación como persona y profesional.

6. COMPETENCES

d) An ability to function on multidisciplinary teams. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

l) Develop principles research in the area of computing with levels of international competitiveness. (**Familiarity**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: (12)	
Competences Expected: C19, C24	
Topics	**Learning Outcomes**
• El subrayado. • Toma de puntes. • La vocación, hábitos de la vida universitaria. • Interacción humana. • La voluntad como requisito para el aprendizaje. • La plantificación y el tiempo	• Analizar la documentación normativa de la Universidad valorando su importancia para la convivencia y desempeño académico. [Usage] • Comprender y valorar la exigencia de la vida universitaria como parte de la formación personal y profesional.[Usage] • Planificar adecuadamente el tiempo en función de sus metas personales y académicas.[Usage] • Elaborar un plan de mejora personal a partir del conocimiento de sí mismo.[Usage]
Readings : [**bibliografiaTecnologia**]	

Unit 2: (12)	
Competences Expected: C19,C24	
Topics	**Learning Outcomes**
• Resumen. Notas al margen. Nemotecnias. • Procesos mentales: Simples, complejos. Fundamentos del aprendizaje significativo. • Los pasos o factores para el aprendizaje. Leyes del aprendizaje. Cuestionario de estilos de aprendizaje Identificación del estilo de aprendizaje personal • La lectura académica. Niveles de análisis de un texto: idea central, idea principal e ideas secundarias. El modelo de Meza de Vernet. • Exámenes: Preparación. Pautas y estrategias para antes, durante y después de un examen. Inteligencia emocional y exámenes. • Las fuentes de información. Aparato crítico: concepto y finalidad. Normas Vancouver. Referencias y citas.	• Identificar los procesos mentales relacionándolos con el aprendizaje [Usage]. • Comprender el proceso del aprendizaje para determinar el estilo propio e incorporarlo en su actividad académica [Usage]. • Desarrollar estrategias para el análisis de textos potenciando la comprensión lectora [Usage]. • Diseñar un programa estratégico para afrontar con éxito los exámenes[Usage].
Readings : [Rod07], [Per10], [Qui07]	

Unit 3: (12)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• Los mapas conceptuales. Características y elementos. • Los derechos de autor y el plagio. Derechos personales o morales. Derechos patrimoniales. “Copyrigth”. • Autoestima, Inteligencia Emocional, Asertividad y Resiliencia. Conceptos, desarrollo y fortalecimiento. • Aparato crítico: Normas Vancouver. Aplicación práctica. • Generación de ideas. Estrategias para organizar las ideas, redacción y revisión.	• Aplicar las técnicas de estudio atendiendo a sus particularidades y adecuándolas a las distintas situaciones que demanda el aprendizaje [Usage]. • Reconocer la importancia del respeto a la propiedad Intelectual [Usage]. • Reconocer la importancia de la Inteligencia Emocional, la conducta asertiva, la autoestima y la resiliencia valorándolas como fortalezas para el desempeño universitario [Usage].
Readings : [Chá11], [Vel99]	

Unit 4: (12)	
Competences Expected: C19	
Topics	**Learning Outcomes**
• Cuadro Sinóptico. Los mapas mentales. Practicas con la temática del curso. • El método personal de estudio. • El aprendizaje cooperativo: definición, los grupos de estudio, organización, roles de los miembros. • Pautas para conformar grupos eficientes y armónicos. • El método personal de estudio.Reforzamiento de técnicas de estudio. • Presentación y exposición de trabajos de producción intelectual. • El debate y la argumentación.	• Aplicar las técnicas de estudio atendiendo a sus particularidades y adecuándolas a las distintas situaciones que demanda el aprendizaje [Usage]. • Asumir el manejo de conductas y actitudes para el aprendizaje cooperativo y el desempeño en los equipos de trabajo [Usage]. • Formular un proyecto de método personal de estudio, de acuerdo a su estilo y necesidades, que incluya técnicas y estrategias [Usage].
Readings : [Rod07], [Chá11]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Chá11] A. Chávez. *Se necesita un tutor*. UCSP, 2011.

[Per10] A.E. Perez. *Teoría del Derecho*. Editorial Madrid, 2010.

[Qui07] V. Quintana. *El estudio Universitario y elementos de investigación científica*. Editorial universitaria, 2007.

[Rod07] J. Rodríguez. *Guía para el método de estudio universitario*. Educa, 2007.

[Vel99] Marco Flores Velazco. *Mapas conceptuales en el aula*. Ed. San Marcos, 1999.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

ID101. Technical and professional English (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	-
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	None

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of people's lives. In the case of foreign languages, English is undoubtedly the most pratical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services English is perhaps the most important practical tool that the student must master from the outset as part of his comprehensive education.

5. GOALS

- Know the English language and its grammatical structure.
- Identify situations and employ dialogues related to them.

6. COMPETENCES

f) An ability to communicate effectively. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Hello everybody! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Verb To Be. • Affirmative sentences , Negatives and Questions. • Number Expressions. • Objects and Countries. • Expressions to greet and make presentations.	• At the end of the first unit, each student, understanding the grammar of the present tense is able to express a greater quantity of expressions of time and also to use sentences with the verb To Be to express situation and state. • That the student is able to analyze and express ideas about dates and numbers in order.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Meeting people! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Possessive adjectives. • Expressions to find prices. • Possession expressions • Vocabulary of Family, Food and Drinks. • Formal requests. • Informal letters.	• At the end of the second unit, students having identified how to express orders and make offerings in restaurants use them in various situations. Explain and apply food and drink vocabulary.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: The world of work (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Simple present tense. Auxiliaries. • Affirmative sentences, Negatives and Questions. • Common verbs and occupations. • Indications for expressing the time.	• At the end of the third unit, students having recognized the characteristics of the present simple, use it to make descriptions of various types. Describe people and places and give directions. Express time.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Take it easy! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Present Simple 2. • Affirmative sentences , Negatives and Questions. • Use of Verbs of entertainment. • Free time. • The seasons of the year. • Expressions of social activities.	• At the conclusion of the fourth unit, the students having identified the idea of expressing ideas of free time actions in Simple and Continuous Present. Express ideas of stations and related activities.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Where do you live? (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Use There is/There are. • Sentences with Prepositions. • Expressions of Quantity. • Vocabulary of airplanes and places. • Expressions of direction indications.	• At the end of the fifth unit, students, based on the understanding of the present continuous time, will elaborate sentences using ideas of location and place. They will also assimilate the need to express objects in common use. They will acquire vocabulary to describe the parts of a house using expressions to ask for directions.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: Can you speak English? (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Can/cant. • Past of verb to be. Use of Could • Vocabulary of Countries and languages. • Expressions for using the phone • Writing formal letters. • Readings.	• At the end of the sixth unit, students having learned the fundamentals of using auxiliary mode, will create sentences applied to the appropriate context. They emphasize the difference between languages and nationalities. They describe feelings. Use expressions on the phone.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: Then and now! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Past Simple. • Expressions of past tense. • Vocabulary regular and irregular verbs • Expressions to describe the climate. • Writing descriptive paragraphs. • Special occasions.	• At the end of the seventh unit, students having learned the basics of structuring the Simple Past experience the need to be able to express this type of time in actions. They will practice in appropriate contexts. They emphasize the difference between irregular and regular verbs. They describe actions with several verbs. They use expressions to describe the climate.
Readings : [SJ02], [Cam06], [Mac99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS112. Computer Science I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	5
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS111. Introduction to Computer Science. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the second course in the sequence of introductory courses in computer science. The course will introduce students in the various topics of the area of computing such as: Algorithms, Data Structures, Software Engineering, etc.

5. GOALS

- Introduce the student to the foundations of the object orientation paradigm, allowing the assimilation of concepts necessary to develop information systems.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

a10) (10)

a11) (11)

b1) (1)

d1) (1)

7. TOPICS

Unit 1: General overwiew of Programming Languages (1)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Brief review of programming paradigms. • Comparison between functional programming and imperative programming. • History of programming languages.	• Discuss the historical context for several programming language paradigms [Familiarity]
Readings : [Str13], [Dei17]	

Unit 2: Virtual Machines (1)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• The virtual machine concept. • Types of virtualization (including Hardware/Software, OS, Server, Service, Network). • Intermediate languages.	• Explain the concept of virtual memory and how it is realized in hardware and software [Familiarity] • Differentiate emulation and isolation [Familiarity] • Evaluate virtualization trade-offs [Assessment]
Readings : [Str13], [Dei17]	

<table>
<tr><td colspan="2"><b>Unit 3: Basic Type Systems (2)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: a,b,i</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• A type as a set of values together with a set of operations
– Primitive types (e.g., numbers, Booleans)
– Compound types built from other types (e.g., records, unions, arrays, lists, functions, references)
• Model statement (link, visibility, scope and life time).
• General view of type checking.</td><td>• For both a primitive and a compound type, informally describe the values that have that type [Familiarity]
• For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Familiarity]
• Describe examples of program errors detected by a type system [Familiarity]
• For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage]
• Give an example program that does not type-check in a particular language and yet would have no error if run [Familiarity]
• Use types and type-error messages to write and debug programs [Usage]
• Explain how typing rules define the set of operations that are legal for a type [Familiarity]
• Write down the type rules governing the use of a particular compound type [Usage]
• Explain why undecidability requires type systems to conservatively approximate program behavior [Familiarity]
• Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage]
• Discuss the differences among generics, subtyping, and overloading [Familiarity]
• Explain multiple benefits and limitations of static typing in writing, maintaining, and debugging software [Familiarity]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Str13], [Dei17]</td></tr>
</table>

Unit 4: Fundamental Programming Concepts (6)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing	• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Assessment] • Identify and describe uses of primitive data types [Familiarity] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Assessment] • Describe the concept of recursion and give examples of its use [Familiarity] • Identify the base case and the general case of a recursively-defined problem [Assessment]
Readings : [Str13], [Dei17]	

Unit 5: Object-Oriented Programming (10)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Object-oriented design – Decomposition into objects carrying state and having behavior – Class-hierarchy design for modeling • Object-oriented idioms for encapsulation – Privacy and visibility of class members – Interfaces revealing only method signatures – Abstract base classes • Definition of classes: fields, methods, and constructors • Subclasses, inheritance, and method overriding • Subtyping – Subtype polymorphism; implicit upcasts in typed languages – Notion of behavioral replacement: subtypes acting like supertypes – Relationship between subtyping and inheritance • Using collection classes, iterators, and other common library components • Dynamic dispatch: definition of method-call	• Design and implement a class [Usage] • Use subclassing to design simple class hierarchies that allow code to be reused for distinct subclasses [Usage] • Correctly reason about control flow in a program using dynamic dispatch [Usage] • Compare and contrast (1) the procedural/functional approach—defining a function for each operation with the function body providing a case for each data variant—and (2) the object-oriented approach—defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Assessment] • Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Familiarity] • Use object-oriented encapsulation mechanisms such as interfaces and private members [Usage] • Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]
Readings : [Str13], [Dei17]	

Unit 6: Algorithms and Design (3)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • The role of algorithms in the problem-solving process • Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • Fundamental design concepts and principles – Abstraction – Program decomposition – Encapsulation and information hiding – Separation of behavior and implementation	• Discuss the importance of algorithms in the problem-solving process [Familiarity] • Discuss how a problem may be solved by multiple algorithms, each with different properties [Familiarity] • Create algorithms for solving simple problems [Usage] • Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage] • Implement, test, and debug simple recursive functions and procedures [Usage] • Determine whether a recursive or iterative solution is most appropriate for a problem [Assessment] • Implement a divide-and-conquer algorithm for solving a problem [Usage] • Apply the techniques of decomposition to break a program into smaller pieces [Usage] • Identify the data components and behaviors of multiple abstract data types [Usage] • Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage] • Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Assessment]
Readings : [Str13], [Dei17]	

Unit 7: Algorithmic Strategies (3)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Brute-force algorithms • Greedy algorithms • Divide-and-conquer • Recursive backtracking • Dynamic Programming	• For each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply [Familiarity] • Use a greedy approach to solve an appropriate problem and determine if the greedy rule chosen leads to an optimal solution [Assessment] • Use a divide-and-conquer algorithm to solve an appropriate problem [Usage] • Use recursive backtracking to solve a problem such as navigating a maze [Usage] • Use dynamic programming to solve an appropriate problem [Usage] • Determine an appropriate algorithmic approach to a problem [Assessment] • Describe various heuristic problem-solving methods [Familiarity]
Readings : [Str13], [Dei17]	

Unit 8: Basic Analysis (2)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm	• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Familiarity]
Readings : [Str13], [Dei17]	

Unit 9: Fundamental Data Structures and Algorithms (6)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort)	• Implement basic numerical algorithms [Usage] • Implement simple search algorithms and explain the differences in their time complexities [Assessment] • Be able to implement common quadratic and O(N log N) sorting algorithms [Usage] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Familiarity] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity] • Explain how tree balance affects the efficiency of various binary search tree operations [Familiarity] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment] • Trace and/or implement a string-matching algorithm [Usage]
Readings : [Str13], [Dei17]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Dei17] Deitel & Deitel. *C++17 - The Complete Guide*. 10th. Pearson, 2017. ISBN: 978-0201734843.

[Str13] Bjarne Stroustrup. *The C++ Programming Language*. 4th. Addison-Wesley, 2013. ISBN: 978-0-321-56384-2.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS1D2. Discrete Structures II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS1D1. Discrete Structures I. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

In order to understand the advanced computational techniques, the students must have a strong knowledge of the Various discrete structures, structures that will be implemented and used in the laboratory in the programming language..

5. GOALS

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student applies efficient travel strategies to be able to search data in an optimal way.
- That the student uses the various counting techniques to solve computational problems.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Familiarity**)

a3) (**3**)

a9) (**9**)

a13) (**13**)

a14) (**14**)

j1) (**1**)

j2) (**2**)

7. TOPICS

<table>
<tr><td colspan="2"><b>Unit 1: Digital Logic and Data Representation (10)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: a,b,i</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• Reticles: Types and properties.
• Boolean algebras.
• Boolean Functions and Expressions.
• Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form.
• Logical gates.
• Circuit Minimization.</td><td>• Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Assessment].
• Explain the algebraic structures of reticulum and its types [Assessment].
• Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Assessment].
• Explain the properties that satisfies a Boolean algebra [Assessment].
• Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Assessment].
• Find the canonical forms of a Boolean function [Assessment].
• Represent a Boolean function as a Boolean circuit using logic gates [Assessment].
• Minimize a Boolean function. [Assessment].</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Ros07], [Gri03]</td></tr>
</table>

Unit 2: Basics of Counting (40)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Counting arguments – Set cardinality and counting – Sum and product rule – Inclusion-exclusion principle – Arithmetic and geometric progressions • The pigeonhole principle • Permutations and combinations – Basic definitions – Pascal's identity – The binomial theorem • Solving recurrence relations – An example of a simple recurrence relation, such as Fibonacci numbers – Other examples, showing a variety of solutions • Basic modular arithmetic	• Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions [Familiarity] • Apply the pigeonhole principle in the context of a formal proof [Familiarity] • Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application [Familiarity] • Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (eg, a full house) [Familiarity] • Solve a variety of basic recurrence relations [Familiarity] • Analyze a problem to determine underlying recurrence relations [Familiarity] • Perform computations involving modular arithmetic [Familiarity]
Readings : [Gri97]	

Unit 3: Graphs and Trees (40)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Trees – Properties – Traversal strategies • Undirected graphs • Directed graphs • Weighted graphs • Spanning trees/forests • Graph isomorphism	• Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Familiarity] • Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Familiarity] • Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Familiarity] • Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Familiarity] • Explain how to construct a spanning tree of a graph [Familiarity] • Determine if two graphs are isomorphic [Familiarity]
Readings : [Joh99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction.* 5 ed. Pearson, 2003.

[Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria.* Addison Wesley Iberoamericana, 1997.

[Joh99] Richard Johnsonbaugh. *Matemáticas Discretas.* Prentice Hall, México, 1999.

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications.* 7 ed. Mc Graw Hill, 2007.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

MA101. Math II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA100. Mathematics I. (1^{st} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The course develops in students the skills to deal with models of science and engineering skills. In the first part of the course a study of the functions of several variables, partial derivatives, multiple integrals and an introduction to vector fields is performed. Then the student will use the basic concepts of calculus to model and solve ordinary differential equations using techniques such as Laplace transforms and Fourier series.

5. GOALS

- Apply derivation rules and partial differentation in functions of several variables.

- Apply techniques for calculating multiple integrals.

- Understand and use the concepts of vector calculus.

- Understand the importance of series.

- Identify and solve differential equations of the first order and their applications in chemical and physical problems.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Multi-Variable Function Differential (24)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Concept of multi-variable functions. • Directional Derivates • Tangent line, normal plane to curve line and tangent plane, normal line to a curve plan. Know to calculate their equations. • Concept of extreme value and conditional extreme value of multi-variable functions • Applications problems such as modeling total production of an economic system, speed of sound through the ocean, thickener optimization, etc.	• Understand the concept of multi-variable functions. • Master the concept and calculation method of the direction derivative and gradient of the guide. • Master the calculation method of the first order and second order partial derivative of composite functions. • Master the calculation method of the partial derivatives for implicit functions. • Understand tangent line, normal plane to curve line and tangent plane, normal line to a curve plan. Know to calculate their equations. • Learn the concept of extreme value and conditional extreme value of multi-variable functions; know to find out the binary function extreme value. • Be able to solve simple applications problems.
Readings : [Ste12], [Zil13]	

Unit 2: Multi-Variable function Integral (12)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Double integral, triple integral and nature of the multiple integral. • Method of double integral • Line Integral • The Divergence, Rotation and Laplacian	• Understand the double integral, triple integral, and understand the nature of the multiple integral. • Master the calculation method of double integral (Cartesian coordinates, polar coordinates) the triple integral (Cartesian coordinates, cylindrical coordinates, spherical coordinates). • Understand the concept of line Integral, their properties and relationships. • Know to calculate the line integral. • Master the calculation the rotational, divergence and Laplacian.
Readings : [Ste12], [Zil13]	

Unit 3: Series (24)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Convergent series • Taylor and McLaurin series • Orthogonal functions	• Master to calculation if series is convergent, and if convergent, find the sum of the series trying to find the radius of convergence and the interval of convergence of a power series. • Represent a function as a power series and find the Taylor and McLaurin Series to estimate function values to a desired accuracy. • Understand the concepts of orthogonal functions and the expansion of a given function f to find its Fourier series.
Readings : [Ste12], [Zil13]	

Unit 4: Ordinary Differential Equations (30)	
Competences Expected: C1,C20	
Topics	**Learning Outcomes**
• Concept of differential equations • Methods to resolve differential equations • Methods to resolve the secod order linear differential equations • Higher order linear ordinary differential equations • Applications problems using Laplace transforms	• Understand differential equations, solutions, order, general solution, initial conditions and special solutions etc. • Master the calculation method for variables separable equation and first order linear equations. Known to solve homogeneous equation and Bernoulli (Bernoulli) equations; understand variable substitution to solve the equation. • Master to solve total differential equations. • Be able to use reduced order method to solve equations. • Understand the structure of the second order linear differential equation. • Master calculation method for the constant coefficient homogeneous linear differential equations; and understand calculation method for the higher order homogeneous linear differential equations. • Know to apply the differential equation calculation method to solve simple geometric and physic application problems. • Solve properly certain types of differential equations using Laplace transforms.
Readings : [Ste12], [Zil13]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Ste12] James Stewart. *Calculus*. 7th. CENGAGE Learning, 2012.

[Zil13] Dennis G. Zill. *Differential equations with Boundary value problems*. 8th. CENGAGE Learning, 2013.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

FG106. Theater (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	FG101. Communication. (1st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Favorece al estudiante a identificarse a la "Comunidad Académica" de la Universidad, en la medida en que le brinda canales naturales de integración a su grupo y a su Centro de Estudios y le permite, desde una visión alternativa, visualizar la valía interior de las personas a su alrededor, a la vez que puede conocer mejor la suya propia. Relaciona al universitario, a través de la experimentación, con un nuevo lenguaje, un medio de comunicación y expresión que va más allá de la expresión verbal conceptualizada. Coadyuva al estudiante en su formación integral, desarrollando en él capacidades corporales. Estimula en él, actitudes anímicas positivas, aptitudes cognitivas y afectivas. Enriquece su sensibilidad y despierta su solidaridad. Desinhibe y socializa, relaja y alegra, abriendo un camino de apertura de conocimiento del propio ser y el ser de los demás.

5. GOALS

- Contribuir a la formación personal y profesional del estudiante, reconociendo, valorando y desarrollando su lenguaje corporal, integrándolo a su grupo, afianzando su seguridad personal, enriqueciendo su intuición, su imaginación y creatividad, motivándolo a abrir caminos de búsqueda de conocimiento de sí mismo y de comunicación con los demás a través de su sensibilidad, de ejercicios de introspección y de nuevas vías de expresión.

6. COMPETENCES

f) An ability to communicate effectively. (**Usage**)

ñ) Understand that the formation of a good professional is not disconnected or opposed but rather contributes to genuine personal growth. This requires the assimilation of solid values, broad spiritual horizons and a deep vision of the cultural environment. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: (6)	
Competences Expected: C18,C24	
Topics	**Learning Outcomes**
• ¿Qué es el Arte? Una experiencia vivencial y personal. • La llave maestra: la creatividad. • La importancia del teatro en la formación personal y profesional. • Utilidad y enfoque del arte teatral.	• Reconocer la vigencia del Arte y la creatividad en el desarrollo personal y social [Usage]. • Relacionar al estudiante con su grupo valorando la importancia de la comunicación humana y del colectivo social [Usage]. • Reconocer nociones básicas del teatro [Usage].
Readings : [Maj58], [Pav98]	

Unit 2: (6)	
Competences Expected: C17,C24	
Topics	**Learning Outcomes**
• Juego, luego existo. • El juego del niño y el juego dramático. • Juegos de integración grupal y juegos de creatividad. • La secuencia teatral.	• Reconocer el juego como herramienta fundamental del teatro [Usage]. • Interiorizar y revalorar el juego como aprendizaje creativo [Usage]. • Acercar al estudiante de manera espontánea y natural, a la vivencia teatral [Usage].
Readings : [Maj58], [Pav98]	

Unit 3: (9)	
Competences Expected: C17, C18, C24	
Topics	**Learning Outcomes**
• Toma de conciencia del cuerpo. • Toma de conciencia del espacio • Toma de conciencia del tiempo • Creación de secuencias individuales y colectivas: Cuerpo, espacio y tiempo. • El uso dramático del elemento: El juego teatral. • Presentaciones teatrales con el uso del elemento.	• Experimentar con nuevas formas de expresión y comunicación [Usage]. • Conocer algunos mecanismos de control y manejo corporal [Usage]. • Brindar caminos para que el alumno pueda desarrollar creativamente su imaginación, su capacidad de relación y captación de estímulos auditivos, rítmicos y visuales [Usage]. • Conocer y desarrollar el manejo de su espacio propio y de sus relaciones espaciales [Usage]. • Experimentar estados emocionales diferentes y climas colectivos nuevos [Usage].
Readings : [Maj58], [Pav98]	

Unit 4: (12)	
Competences Expected: C18, C24	
Topics	**Learning Outcomes**
• Relajación, concentración y respiración. • Desinhibición e interacción con el grupo. • La improvisación. • Equilibrio, peso, tiempo y ritmo. • Análisis del movimiento. Tipos de movimiento. • La presencia teatral. • La danza, la coreografía teatral.	• Ejercitarse en el manejo de destrezas comunicativas no verbales [Usage]. • Practicar juegos y ejercicios de lenguaje corporal, individual y grupalmente [Usage]. • Expresar libre y creativamente sus emociones y sentimientos y su visión de la sociedad a través de representaciones originales con diversos lenguajes [Usage]. • Conocer los tipos de actuación [Usage].
Readings : [Maj58], [Pav98]	

Unit 5: (3)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• El orígen del teatro, el teatro griego y el teatro romano. • El teatro medieval , la comedia del arte. • De la pasión a la razón: Romanticismo e Ilustración. • El teatro realista, teatro épico. Brech y Stanislavski. • El teatro del absurdo, teatro contemporáneo y teatro total. • Teatro en el Perú: Yuyashkani, La Tarumba, pataclaun, otros.	• Conocer la influencia que la sociedad ha ejercido en el teatro y la respuesta de este arte ante los diferentes momentos de la historia [Usage]. • Apreciar el valor y aporte de las obras de dramaturgos importantes [Usage]. • Analizar el contexto social del arte teatral [Usage]. • Reflexionar sobre el Teatro Peruano y arequipeño [Usage].
Readings : [Maj58], [Pav98]	

Unit 6: (12)	
Competences Expected: C17,C18, C24	
Topics	**Learning Outcomes**
• Apreciación teatral. Expectación de una o más obras teatrales. • El espacio escénico. • Construcción del personaje • Creación y montaje de una obra teatral . • Presentación en público de pequeñas obras haciendo uso de vestuario, maquillaje, escenografía, utilería y del empleo dramático del objeto.	• Emplear la creación teatral, como manifestación de ideas y sentimientos propios ante la sociedad [Usage]. • Aplicar las técnicas practicadas y los conocimientos aprendidos en una apreciación y/o expresión teatral concreta que vincule el rol de la educación [Usage]. • Intercambiar experiencias y realizar presentaciones breves de ejercicios teatrales en grupo, frente a público [Usage].
Readings : [Maj58], [Pav98]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Maj58] Angel Majorana. *El arte de hablar en publico*. La España Moderna, 1958.

[Pav98] Patrice Pavis. *Diccionario del Teatro*. Edit. Piados BA, 1998.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

ID102. Technical and professional English II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	6
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	ID101. Technical and professional English. (1st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of people's lives. In the case of foreign languages, English is undoubtedly the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services english is perhaps the most important practical tool that the student must master from the outset as part of his comprehensive education.

5. GOALS

- Develop the ability to fluently speak the language..
- Increase vocabulary and handle simple expressions

6. COMPETENCES

f) An ability to communicate effectively. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: How long ago? (0)	
Competences Expected: 2	
Topics	**Learning Outcomes**
• Simple past • Negative sentences with ago. • Conjunctions • Expressions of Time in past • Phonetic relations and symbols • Expressions to give the date	• At the end of the eighth unit, each student, understanding the grammar of the past tense is able to express a greater number of expressions of time and also to use prepositions to describe varied places and times. It is also capable of analyzing and expressing ideas about dates and numbers in order.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Food you like! (0)	
Competences Expected: 2	
Topics	**Learning Outcomes**
• Accounting and Non-Accounting Nouns • Expressions with Would like and I'd like • Quantifiers • Meals around the world • Formal requests • Formal letters	• At the end of the ninth unit, students having identified how to express orders and make collections, uses in various situations. Express situations and states related to quantities. Explain and apply food and drinks vocabulary.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: The world of work (0)	
Competences Expected: 2	
Topics	**Learning Outcomes**
• Adjectives • Sentences with Comparative Adjectives • Sentences with Superlative Adjectives • Cities and countryside • Directional indications	• At the end of the tenth unit, students who have recognized the characteristics of adjectives use these to make comparisons of various types. Describes people and places and directions. They will use conjunctions to unite type ideas.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Looking good! (0)	
Competences Expected: 2	
Topics	**Learning Outcomes**
• Present continuous • Affirmative sentences, Negatives and Questions • Use of Whose • Possessive pronouns • Clothing and colors • Expressions to wear in clothing stores • Phonetic symbols.	• At the end of the eleventh unit, students having identified the idea of expressing ideas of actions that occur at the time or that relate at any time structure sentences in Present Progressive. They express ideas of possession with regard to clothes and colors.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Life is an adventure! (0)	
Competences Expected: 2	
Topics	**Learning Outcomes**
• Use of going to • Future time sentences • Expressions of Quantity. • Action verbs • Vocabulary of the climate • Expressions of Suggestion • Write a postcard	• At the end of the twelfth unit, students, from the understanding of future time, will elaborate sentences using the necessary elements. They will also assimilate the need to express purposeful infinitives. They will acquire vocabulary to describe the climate. Expressions will be presented to make and ask for suggestions.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: You‘re pretty smart! (0)	
Competences Expected: 2	
Topics	**Learning Outcomes**
• Forms of Questions • Adverbs and Adjectives • Vocabulary description of feelings • Expressions for train travel • Writing Short Stories • Readings	• At the end of the thirteenth unit, students having learned the fundamentals of structuring various questions, will perform application work in appropriate contexts. They emphasize the difference between adjectives and adverbs. They describe feelings. They use expressions to catch a train. They assume the idea is suffixes and prefixes.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: Have you ever? (0)	
Competences Expected: 2	
Topics	**Learning Outcomes**
• Perfect present • Keywords with never, ever, and yet • Vocabulary verbs in Past participle • Expressions for airplane travel • Writing thank-you letters • Readings	• At the end of the fourteenth unit, students having known the fundamentals of the structure of the Present Perfect experience the need to express this type of time in actions. They will practice in appropriate contexts. They emphasize the difference between simple past and perfect present. Describe actions with never, ever, and yet. They use expressions to use at an airport.
Readings : [SJ02], [Cam06], [Mac99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 1 Student Book*. Editorial Oxford, 2002.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS113. Computer Science II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS112. Computer Science I. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This is the third course in the sequence of introductory courses in computer science. This course is intended to cover Concepts indicated by the Computing Curriculum IEEE (c) -ACM 2001, under the functional-first approach. The object-oriented paradigm allows us to combat complexity by making models from abstractions of the problem elements and using techniques such as encapsulation, modularity, polymorphism and inheritance. The Dominion of these topics will enable participants to provide computational solutions to design problems simple of the real world.

5. GOALS

- Introduce the student in the fundaments of the paradigm of object orientation, allowing the assimilation of concepts necessary to develop an information system

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

a10) (**10**)

a11) (**11**)

b1) (**1**)

d1) (**1**)

7. TOPICS

Unit 1: Fundamental Programming Concepts (5)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Basic syntax and semantics of a higher-level language • Variables and primitive data types (e.g., numbers, characters, Booleans) • Expressions and assingments • Simple I/O including file I/O • Conditional and iterative control structures • Functions and parameter passing • The concept of recursion	• Analyze and explain the behavior of simple programs involving the fundamental programming constructs variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. [Usage] • Identify and describe uses of primitive data types [Usage] • Write programs that use primitive data types [Usage] • Modify and expand short programs that use standard conditional and iterative control structures and functions [Usage] • Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing [Usage] • Write a program that uses file I/O to provide persistence across multiple executions [Usage] • Choose appropriate conditional and iteration constructs for a given programming task [Usage] • Describe the concept of recursion and give examples of its use [Usage] • Identify the base case and the general case of a recursively-defined problem [Usage]
Readings : [**stroustrup2013**], [Van02], [LE13]	

Unit 2: Object-Oriented Programming (7)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Object-oriented design – Decomposition into objects carrying state and having behavior – Class-hierarchy design for modeling • Definition of classes: fields, methods, and constructors • Subclasses, inheritance, and method overriding • Dynamic dispatch: definition of method-call • Subtyping – Subtype polymorphism; implicit upcasts in typed languages – Notion of behavioral replacement: subtypes acting like supertypes – Relationship between subtyping and inheritance • Object-oriented idioms for encapsulation – Privacy and visibility of class members – Interfaces revealing only method signatures – Abstract base classes • Using collection classes, iterators, and other common library components	• Design and implement a class [Usage] • Use subclassing to design simple class hierarchies that allow code to be reused for distinct subclasses [Usage] • Correctly reason about control flow in a program using dynamic dispatch [Usage] • Compare and contrast (1) the procedural/functional approach—defining a function for each operation with the function body providing a case for each data variant—and (2) the object-oriented approach—defining a class for each data variant with the class definition providing a method for each operation Understand both as defining a matrix of operations and variants [Usage] • Explain the relationship between object-oriented inheritance (code-sharing and overriding) and subtyping (the idea of a subtype being usable in a context that expects the supertype) [Usage] • Use object-oriented encapsulation mechanisms such as interfaces and private members [Usage] • Define and use iterators and other operations on aggregates, including operations that take functions as arguments, in multiple programming languages, selecting the most natural idioms for each language [Usage]
Readings : [stroustrup2013]	

Unit 3: Algorithms and Design (5)	
Competences Expected: a,b,d	
Topics	**Learning Outcomes**
• The concept and properties of algorithms – Informal comparison of algorithm efficiency (e.g., operation counts) • The role of algorithms in the problem-solving process • Problem-solving strategies – Iterative and recursive mathematical functions – Iterative and recursive traversal of data structures – Divide-and-conquer strategies • Fundamental design concepts and principles – Abstraction – Program decomposition – Encapsulation and information hiding – Separation of behavior and implementation	• Discuss the importance of algorithms in the problem-solving process [Usage] • Discuss how a problem may be solved by multiple algorithms, each with different properties [Usage] • Create algorithms for solving simple problems [Usage] • Use a programming language to implement, test, and debug algorithms for solving simple problems [Usage] • Implement, test, and debug simple recursive functions and procedures [Usage] • Determine whether a recursive or iterative solution is most appropriate for a problem [Usage] • Implement a divide-and-conquer algorithm for solving a problem [Usage] • Apply the techniques of decomposition to break a program into smaller pieces [Usage] • Identify the data components and behaviors of multiple abstract data types [Usage] • Implement a coherent abstract data type, with loose coupling between components and behaviors [Usage] • Identify the relative strengths and weaknesses among multiple designs or implementations for a problem [Usage]
Readings : [**stroustrup2013**], [**Weert16**], [LE13]	

Unit 4: Basic Analysis (3)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm • Asymptotic analysis of upper and expected complexity bounds • Big O notation: formal definition • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Empirical measurements of performance • Time and space trade-offs in algorithms • Big O notation: use • Little o, big omega and big theta notation • Recurrence relations • Analysis of iterative and recursive algorithms • Master Theorem and Recursion Trees	• Explain what is meant by "best", "expected", and "worst" case behavior of an algorithm [Usage] • In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors [Usage] • Determine informally the time and space complexity of different algorithms [Usage] • State the formal definition of big O [Usage] • List and contrast standard complexity classes [Usage] • Perform empirical studies to validate hypotheses about runtime stemming from mathematical analysis Run algorithms on input of various sizes and compare performance [Usage] • Give examples that illustrate time-space trade-offs of algorithms [Usage] • Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms [Usage] • Use big O notation formally to give expected case bounds on time complexity of algorithms [Usage] • Explain the use of big omega, big theta, and little o notation to describe the amount of work done by an algorithm [Usage] • Use recurrence relations to determine the time complexity of recursively defined algorithms [Usage] • Solve elementary recurrence relations, eg, using some form of a Master Theorem [Usage]
Readings : [stroustrup2013]	

Unit 5: Basic Type Systems (5)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• A type as a set of values together with a set of operations – Primitive types (e.g., numbers, Booleans) – Compound types built from other types (e.g., records, unions, arrays, lists, functions, references) • Association of types to variables, arguments, results, and fields • Type safety and errors caused by using values inconsistently given their intended types • Goals and limitations of static typing – Eliminating some classes of errors without running the program – Undecidability means static analysis must conservatively approximate program behavior • Generic types (parametric polymorphism) – Definition – Use for generic libraries such as collections – Comparison with ad hoc polymorphism (overloading) and subtype polymorphism • Complementary benefits of static and dynamic typing – Errors early vs. errors late/avoided – Enforce invariants during code development and code maintenance vs. postpone typing decisions while prototyping and conveniently allow flexible coding patterns such as heterogeneous collections – Avoid misuse of code vs. allow more code reuse – Detect incomplete programs vs. allow incomplete programs to run	• For both a primitive and a compound type, informally describe the values that have that type [Usage] • For a language with a static type system, describe the operations that are forbidden statically, such as passing the wrong type of value to a function or method [Usage] • Describe examples of program errors detected by a type system [Usage] • For multiple programming languages, identify program properties checked statically and program properties checked dynamically [Usage] • Give an example program that does not type-check in a particular language and yet would have no error if run [Usage] • Use types and type-error messages to write and debug programs [Usage] • Explain how typing rules define the set of operations that are legal for a type [Usage] • Write down the type rules governing the use of a particular compound type [Usage] • Explain why undecidability requires type systems to conservatively approximate program behavior [Usage] • Define and use program pieces (such as functions, classes, methods) that use generic types, including for collections [Usage] • Discuss the differences among generics, subtyping, and overloading [Usage] • Explain multiple benefits and limitations of static typing in writing, maintaining, and debugging software [Usage]
Readings : [**stroustrup2013**]	

Unit 6: Fundamental Data Structures and Algorithms (3)	
Competences Expected: a,b,d	
Topics	**Learning Outcomes**
• Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, • Sequential and binary search algorithms • Worst case quadratic sorting algorithms (selection, insertion) • Worst or average case O(N log N) sorting algorithms (quicksort, heapsort, mergesort) • Hash tables, including strategies for avoiding and resolving collisions • Binary search trees – Common operations on binary search trees such as select min, max, insert, delete, iterate over tree • Graphs and graph algorithms – Representations of graphs (e.g., adjacency list, adjacency matrix) – Depth- and breadth-first traversals • Heaps • Graphs and graph algorithms – Maximum and minimum cut problem – Local search • Pattern matching and string/text algorithms (e.g., substring matching, regular expression matching, longest common subsequence algorithms)	• Implement basic numerical algorithms [Usage] • Implement simple search algorithms and explain the differences in their time complexities [Usage] • Be able to implement common quadratic and O(N log N) sorting algorithms [Usage] • Describe the implementation of hash tables, including collision avoidance and resolution [Usage] • Discuss the runtime and memory efficiency of principal algorithms for sorting, searching, and hashing [Usage] • Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Usage] • Explain how tree balance affects the efficiency of various binary search tree operations [Usage] • Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Usage] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Usage] • Describe the heap property and the use of heaps as an implementation of priority queues [Usage] • Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Usage] • Trace and/or implement a string-matching algorithm [Usage]
Readings : [**stroustrup2013**], [PA18]	

Unit 7: Event-Driven and Reactive Programming (2)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Events and event handlers • Canonical uses such as GUIs, mobile devices, robots, servers • Using a reactive framework – Defining event handlers/listeners – Main event loop not under event-handler-writer's control • Externally-generated events and program-generated events • Separation of model, view, and controller	• Write event handlers for use in reactive systems, such as GUIs [Usage] • Explain why an event-driven programming style is natural in domains where programs react to external events [Usage] • Describe an interactive system in terms of a model, a view, and a controller [Usage]
Readings : [**stroustrup2013**], [Wil11]	

Unit 8: Graphs and Trees (7)	
Competences Expected: a,b,d	
Topics	**Learning Outcomes**
• Trees – Properties – Traversal strategies • Undirected graphs • Directed graphs • Weighted graphs • Spanning trees/forests • Graph isomorphism	• Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Usage] • Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Usage] • Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Usage] • Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Usage] • Explain how to construct a spanning tree of a graph [Usage] • Determine if two graphs are isomorphic [Usage]
Readings : [Nak13]	

Unit 9: Software Design (6)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion , re-use of standard structures • Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented • Structural and behavioral models of software designs • Design patterns • Relationships between requirements and designs: transformation of models, design of contracts, invariants • Software architecture concepts and standard architectures (e.g. client-server, n-layer, transform centered, pipes-and-filters) • The use of component desing: component selection, design, adaptation and assembly of components, component and patterns, components and objects (for example, building a GUI using a standar widget set) • Refactoring designs using design patterns • Internal design qualities, and models for them: efficiency and performance, redundacy and fault tolerance, traceability of requeriments • Measurement and analysis of design quality • Tradeoffs between different aspects of quality • Application frameworks • Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling, transaction processing monitors, workflow systems • Principles of secure design and coding – Principle of least privilege – Principle of fail-safe defaults – Principle of psychological acceptability	• Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation [Usage] • Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design [Usage] • Construct models of the design of a simple software system that are appropriate for the paradigm used to design it [Usage] • Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system [Usage] • For a simple system suitable for a given scenario, discuss and select an appropriate design paradigm [Usage] • Create appropriate models for the structure and behavior of software products from their requirements specifications [Usage] • Explain the relationships between the requirements for a software product and its design, using appropriate models [Usage] • For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system [Usage] • Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server [Usage] • Investigate the impact of software architectures selection on the design of a simple system [Usage] • Apply simple examples of patterns in a software design [Usage] • Describe a form of refactoring and discuss when it may be applicable [Usage] • Select suitable components for use in the design of a software product [Usage] • Explain how suitable components might need to be adapted for use in the design of a software product [Usage] • Design a contract for a typical small software component for use in a given system [Usage] • Discuss and select appropriate software architecture for a simple system suitable for a given scenario [Usage] • Apply models for internal and external qualities in designing software components to achieve an acceptable tradeoff between conflicting quality aspects [Us-

Unit 10: Requirements Engineering (1)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Describing functional requirements using, for example, use cases or users stories • Properties of requirements including consistency, validity, completeness, and feasibility • Software requirements elicitation • Describing system data using, for example, class diagrams or entity-relationship diagrams • Non functional requirements and their relationship to software quality • Evaluation and use of requirements specifications • Requirements analysis modeling techniques • Acceptability of certainty / uncertainty considerations regarding software / system behavior • Prototyping • Basic concepts of formal requirements specification • Requirements specification • Requirements validation • Requirements tracing	• List the key components of a use case or similar description of some behavior that is required for a system [Usage] • Describe how the requirements engineering process supports the elicitation and validation of behavioral requirements [Usage] • Interpret a given requirements model for a simple software system [Usage] • Describe the fundamental challenges of and common techniques used for requirements elicitation [Usage] • List the key components of a data model (eg, class diagrams or ER diagrams) [Usage] • Identify both functional and non-functional requirements in a given requirements specification for a software system [Usage] • Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements [Usage] • Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system [Usage] • Compare the plan-driven and agile approaches to requirements specification and validation and describe the benefits and risks associated with each [Usage] • Use a common, non-formal method to model and specify the requirements for a medium-size software system [Usage] • Translate into natural language a software requirements specification (eg, a software component contract) written in a formal specification language [Usage] • Create a prototype of a software system to mitigate risk in requirements [Usage] • Differentiate between forward and backward tracing and explain their roles in the requirements validation process [Usage]
Readings : [stroustrup2013]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[LE13] Stanley B. Lippman and Barbara E.Moo. *C++ Primer*. 5th. O'Reilly, 2013. ISBN: 9780133053043.

[Nak13] S. Nakariakov. *The Boost C++ Libraries: Generic Programming*. CreateSpace Independent Publishing Platforml, 2013.

[PA18] Praseed Pai and Peter Abraham. *C++ Reactive Programming*. 1st. Packt, 2018.

[Van02] David Vandervoorde. *C++ Templates:The Complete Guide*. 1st. Addison-Wesley, 2002. ISBN: 978-0134448237.

[Wil11] Anthony Williams. *C++ Concurrency in Action*. 1st. Manning, 2011.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS221. Computer Systems Architecture (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS1D2. Discrete Structures II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

A computer scientist must have a solid knowledge of the organization and design principles of diverse computer systems, by understanding the limitations of modern systems they could propose next-gen paradigms. This course teaches the basics and principles of Computer Architecture. This class addreses digital logic design, basics of Computer Architecture and processor design (Instruction Set architecture, microarchitecture, out-of-order execution, branch prediction), execution paradigms (superscalar, dataflow, VLIW, SIMD, GPUs, systolic, multithreading) and memory system organization.

5. GOALS

- Provide a first approach in Computer Architecture.
- Study the design and evolution of computer architectures, which lead to modern approaches and implementations in computing systems.
- Provide fine-grained details of computer hardware, and its relation with software execution.
- Implement a simple microprocessor using Verilog language.

6. COMPETENCES

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Digital logic and digital systems (18)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Overview and history of computer architecture • Combinational and sequential logic/Field programmable gate arrays as a fundamental combinational + sequential logic building block • Abstraction models • Computer-aided design tools that process hardware and architectural representations • Register transfer notation/Hardware Description Language (Verilog/VHDL) • Physical constraints (gate delays, fan-in, fan-out, energy/power)	• Describe the progression of technology devices from vacuum tubes to VLSI, from mainframe computer architectures to the organization of warehouse-scale computers [Familiarity] • Comprehend the trend of modern computer architectures towards multi-core and that parallelism is inherent in all hardware systems [Usage] • Explain the implications of the "power wall" in terms of further processor performance improvements and the drive towards harnessing parallelism [Usage] • Articulate that there are many equivalent representations of computer functionality, including logical expressions and gates, and be able to use mathematical expressions to describe the functions of simple combinational and sequential circuits [Familiarity] • Design the basic building blocks of a computer: arithmetic-logic unit (gate-level), registers (gate-level), central processing unit (register transfer-level), memory (register transfer-level) [Usage] • Use CAD tools for capture, synthesis, and simulation to evaluate simple building blocks (eg, arithmetic-logic unit, registers, movement between registers) of a simple computer design [Familiarity] • Evaluate the functional and timing diagram behavior of a simple processor implemented at the logic circuit level [Assessment]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 2: Machine level representation of data (8)	
Competences Expected: g	
Topics	**Learning Outcomes**
• Bits, bytes, and words • Numeric data representation and number bases • Fixed- and floating-point systems • Signed and twos-complement representations • Representation of non-numeric data (character codes, graphical data) • Representation of registers and arrays	• Explain why everything is data, including instructions, in computers [Assessment] • Explain the reasons for using alternative formats to represent numerical data [Familiarity] • Describe how negative integers are stored in sign-magnitude and twos-complement representations [Usage] • Explain how fixed-length number representations affect accuracy and precision [Usage] • Describe the internal representation of non-numeric data, such as characters, strings, records, and arrays [Usage] • Convert numerical data from one format to another [Usage]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 3: Assembly level machine organization (8)	
Competences Expected: b,g	
Topics	**Learning Outcomes**
• Basic organization of the von Neumann machine • Control unit; instruction fetch, decode, and execution • Instruction sets and types (data manipulation, control, I/O) • Assembly/machine language programming • Instruction formats • Addressing modes • Subroutine call and return mechanisms • I/O and interrupts • Heap vs. Static vs. Stack vs. Code segments	• Explain the organization of the classical von Neumann machine and its major functional units [Familiarity] • Describe how an instruction is executed in a classical von Neumann machine, with extensions for threads, multiprocessor synchronization, and SIMD execution [Familiarity] • Describe instruction level parallelism and hazards, and how they are managed in typical processor pipelines [Familiarity] • Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler [Familiarity] • Demonstrate how to map between high-level language patterns into assembly/machine language notations [Usage] • Explain different instruction formats, such as addresses per instruction and variable length vs fixed length formats [Usage] • Explain how subroutine calls are handled at the assembly level [Usage] • Explain the basic concepts of interrupts and I/O operations [Familiarity] • Write simple assembly language program segments [Usage] • Show how fundamental high-level programming constructs are implemented at the machine-language level [Usage]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 4: Functional organization (8)

Competences Expected: b,g

Topics	**Learning Outcomes**
• Implementation of simple datapaths, including instruction pipelining, hazard detection and resolution • Control unit: microprogrammed • Instruction pipelining • Introduction to instruction-level parallelism (ILP)	• Compare alternative implementation of datapaths [Assessment] • Discuss the concept of control points and the generation of control signals using hardwired or microprogrammed implementations [Familiarity] • Explain basic instruction level parallelism using pipelining and the major hazards that may occur [Usage] • Design and implement a complete processor, including datapath and control [Usage] • Determine, for a given processor and memory system implementation, the average cycles per instruction [Assessment]

Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]

Unit 5: Memory system organization and architecture (8)

Competences Expected: b,g

Topics	**Learning Outcomes**
• Storage systems and their technology • Memory hierarchy: importance of temporal and spatial locality • Main memory organization and operations • Latency, cycle time, bandwidth, and interleaving • Cache memories (address mapping, block size, replacement and store policy) • Multiprocessor cache consistency/Using the memory system for inter-core synchronization/atomic memory operations • Virtual memory (page table, TLB) • Fault handling and reliability • Error coding, data compression, and data integrity	• Identify the main types of memory technology (eg, SRAM, DRAM, Flash, magnetic disk) and their relative cost and performance [Familiarity] • Explain the effect of memory latency on running time [Familiarity] • Describe how the use of memory hierarchy (cache, virtual memory) is used to reduce the effective memory latency [Usage] • Describe the principles of memory management [Usage] • Explain the workings of a system with virtual memory management [Usage] • Compute Average Memory Access Time under a variety of cache and memory configurations and mixes of instruction and data references [Assessment]

Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]

Unit 6: Interfacing and communication (8)	
Competences Expected: b,g,i	
Topics	**Learning Outcomes**
• I/O fundamentals: handshaking, buffering, programmed I/O, interrupt-driven I/O • Interrupt structures: vectored and prioritized, interrupt acknowledgment • External storage, physical organization, and drives • Buses: bus protocols, arbitration, direct-memory access (DMA) • Introduction to networks: communications networks as another layer of remote access • Multimedia support • RAID architectures	• Explain how interrupts are used to implement I/O control and data transfers [Familiarity] • Identify various types of buses in a computer system [Familiarity] • Describe data access from a magnetic disk drive [Usage] • Compare common network organizations, such as ethernet/bus, ring, switched vs routed [Assessment] • Identify the cross-layer interfaces needed for multimedia access and presentation, from image fetch from remote storage, through transport over a communications network, to staging into local memory, and final presentation to a graphical display [Familiarity] • Describe the advantages and limitations of RAID architectures [Familiarity]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 7: Multiprocessing and alternative architectures (8)	
Competences Expected: i	
Topics	**Learning Outcomes**
• Power Law • Example SIMD and MIMD instruction sets and architectures • Interconnection networks (hypercube, shuffle-exchange, mesh, crossbar) • Shared multiprocessor memory systems and memory consistency • Multiprocessor cache coherence	• Discuss the concept of parallel processing beyond the classical von Neumann model [Assessment] • Describe alternative parallel architectures such as SIMD and MIMD [Familiarity] • Explain the concept of interconnection networks and characterize different approaches [Usage] • Discuss the special concerns that multiprocessing systems present with respect to memory management and describe how these are addressed [Familiarity] • Describe the differences between memory backplane, processor memory interconnect, and remote memory via networks, their implications for access latency and impact on program performance [Assessment]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

Unit 8: Performance enhancements (8)	
Competences Expected: g,i	
Topics	**Learning Outcomes**
• Superscalar architecture • Branch prediction, Speculative execution, Out-of-order execution • Prefetching • Vector processors and GPUs • Hardware support for multithreading • Scalability • Alternative architectures, such as VLIW/EPIC, and Accelerators and other kinds of Special-Purpose Processors	• Describe superscalar architectures and their advantages [Familiarity] • Explain the concept of branch prediction and its utility [Usage] • Characterize the costs and benefits of prefetching [Assessment] • Explain speculative execution and identify the conditions that justify it [Assessment] • Discuss the performance advantages that multithreading offered in an architecture along with the factors that make it difficult to derive maximum benefits from this approach [Assessment] • Describe the relevance of scalability to performance [Assessment]
Readings : [HH12], [PP05], [PH04], [JAs07], [HP06], [Par05], [Sta10], [PCh06]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[HH12] David Harris and Sarah Harris. *Digital Design and Computer Architecture.* 2nd. Morgan Kaufmann, 2012. ISBN: 978-0123944245.

[HP06] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach.* 4th. San Mateo, CA: Morgan Kaufman, 2006.

[JAs07] Peter J.Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog.* Morgan Kaufmann, 2007. ISBN: 978-0123695277.

[Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers.* New York: Oxford Univ. Press, 2005. ISBN: ISBN 0-19-515455-X.

[PCh06] Pong P.Chu. *RTL Hardware Design Using VHDL.* 1st. Wiley-Interscience, 2006.

[PH04] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface.* 3rd ed. San Mateo, CA: Morgan Kaufman, 2004.

[PP05] Yale N Patt and Sanjay J Patel. *Introduction to Computing Systems*. 2nd. McGraw Hill, 2005.

[Sta10] William Stalings. *Computer Organization and Architecture: Designing for Performance*. 8th. Upper Saddle River, NJ: Prentice Hall, 2010.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS2B1. Platform Based Development (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS112. Computer Science I. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The world has changed due to the use of fabric and related technologies, rapid, timely and personalized access to the information, through web technology, ubiquitous and pervasive; they have changed the way we do things, how do we think? and how does the industry develop? Web technologies, ubiquitous and pervasive are based on the development of web services, web applications and mobile applications, which are necessary to understand the architecture, design, and implementation of web services, web applications and mobile applications.

5. GOALS

- That the student is able to design and implement services, web applications using tools and languages such as HTML, CSS, JavaScript (including AJAX), back-end scripting and a database, at an intermediate level.
- That the student is able to develop mobile applications, administration of web servers in a Unix system and an introduction to web security, at an intermediate level.

6. COMPETENCES

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

c3) (**3**)

c4) (4)

c5) (**5**)

c6) (**6**)

c7) (7)

c8) (8)

c9) (9)

c10) (10)

d1) (1)

d2) (2)

g1) (1)

i2) (2)

7. TOPICS

Unit 1: Introduction (5)	
Competences Expected: g	
Topics	**Learning Outcomes**
• Overview of platforms (e.g., Web, Mobile, Game, Industrial) • Programming via platform-specific APIs • Overview of Platform Languages (e.g., Objective C, HTML5) • Programming under platform constraints	• Describe how platform-based development differs from general purpose programming [Familiarity] • List characteristics of platform languages [Familiarity] • Write and execute a simple platform-based program [Familiarity] • List the advantages and disadvantages of programming with platform constraints [Familiarity]
Readings : [**fielding2000fielding**], [**grove2009web**], [**annuzzi2013introduction**], [**Cornez2015**]	

Unit 2: Web Platforms (5)	
Competences Expected: c,g,i	
Topics	**Learning Outcomes**
• Web programming languages (e.g., HTML5, Java Script, PHP, CSS) • • Web Platform constraints: Client-Server, Stateless-Stateful, Cache, Uniform Interface, Layered System, Code on Demand, ReST. • Web platform constraints • Software as a Service (SaaS) • Web standards	• Design and Implement a simple web application [Familiarity] • Describe the constraints that the web puts on developers [Familiarity] • Compare and contrast web programming with general purpose programming [Familiarity] • Describe the differences between Software-as-a-Service and traditional software products [Familiarity] • Discuss how web standards impact software development [Familiarity] • Review an existing web application against a current web standard [Familiarity]
Readings : [**fielding2000fielding**]	

Unit 3: Desarrollo de servicios y aplicaciones web (25)

Competences Expected: c,d,g,i

Topics	**Learning Outcomes**
• Describe, identify and debug issues related to web application development • Design and development of interactive web applications using HTML5 and Python • Use MySQL for data management and manipulate MySQL with Python • Design and development of asynchronous web applications using Ajax techniques • Using dynamic client side Javascript scripting language and server side python scripting language with Ajax • Apply XML / JSON technologies for data management with Ajax • Use framework, services and Ajax web APIs and apply design patterns to web application development	• Server-side python scripting language: variables, data types, operations, strings, functions, control statements, arrays, files and directory access, maintain state. [Usage] • Web programming approach using embedded python. [Usage] • Accessing and Manipulating MySQL. [Usage] • The Ajax web application development approach. [Usage] • DOM and CSS used in JavaScript. [Usage] • Asynchronous Content Update Technologies. [Usage] • XMLHttpRequest objects use to communicate between clients and servers. [Usage] • XML and JSON. [Usage] • XSLT and XPath as mechanisms for transforming XML documents. [Usage] • Web services and APIs (especially Google Maps). [Usage] • Macros Ajax for the development of contemporary web applications. [Usage] • Design patterns used in web applications. [Usage]

Readings : [**freeman2011head**]

Unit 4: Mobile Platforms (5)

Competences Expected: c,d,g,i

Topics	**Learning Outcomes**
• Mobile programming languages • Design Principles: Segregation of Interfaces, Single Responsability, Separation of concerns, Dependency Inversion. • Challenges with mobility and wireless communication • Location-aware applications • Performance / power tradeoffs • Mobile platform constraints • Emerging technologies	• Design and implement a mobile application for a given mobile platform [Familiarity] • Discuss the constraints that mobile platforms put on developers [Familiarity] • Discuss the performance vs power tradeoff [Familiarity] • Compare and Contrast mobile programming with general purpose programming [Familiarity]

Readings : [**martin2017clean**], [**annuzzi2013introduction**]

Unit 5: Mobile Applications for Android Handheld Systems (25)	
Competences Expected: c,d,g,i	
Topics	**Learning Outcomes**
• The Android Platform • The Android Development Environment • Application Fundamentals • The Activity Class • The Intent Class • Permissions • The Fragment Class • User Interface Classes • User Notifications • The BroadcastReceiver Class • Threads, AsyncTask & Handlers • Alarms • Networking (http class) • Multi-touch & Gestures • Sensors • Location & Maps	• Students identify necessary software and install it on their personal computers. • Students perform various tasks to familiarize themselves with the Android platform and Environment for development. [Usage] • Students build applications that trace the lifecycle callback methods emitted by the Android platform and demonstrate the behavior of Android when device configuration changes (for example, when the device moves from vertical to horizontal and vice versa). [Usage] • Students build applications that require starting multiple activities through both standard and custom methods. [Usage] • Students build applications that require standard and custom permissions. [Usage] • Students build an application that uses a single code base, but creates different user interfaces depending on the screen size of a device. [Usage] • Students construct a to-do list manager using the user interface elements discussed in class. The application allows users to create new items and to display them in a ListView. [Usage] • Students build an application that uses location information to collect latitude, length of places they visit. [Usage]
Readings : [**annuzzi2013introduction**], [**Cornez2015**]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

MA203. Statistics and Probabilities (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA100. Mathematics I. (1st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

It provides an introduction to probability theory and statistical inference with applications, needs in data analysis, design of random models and decision making.

5. GOALS

- An ability to design and conduct experiments, as well as to analyze and interpret data.
- An ability to identify, formulate, and solve real problems.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Variable Type (6)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Variable Type: Continuous, discrete	• Classify the relevant variables identified according to their type: continuous (interval and ratio), categorical (nominal, ordinal, dichotomous). • Identify the relevant variables of a system using a process approach.
Readings : [MRo14], [Men14]	

Unit 2: Descriptive Statistics (6)	
Competences Expected: C1	
Topics	**Learning Outcomes**
- Central Tendency (Mean, median, mode) - Dispersion (Range, standard deviation, quartile) - Graphics: histogram, boxplot, etc.: Communication ability.	- Use central tendency measures and dispersion measures to describe the data gathered. - Use graphics to communicate the characteristics of the data gathered.
Readings : [MRo14], [Men14]	

Unit 3: Inferential Statistics (6)	
Competences Expected: CS2	
Topics	**Learning Outcomes**
- Determination of the sample size - Confidence interval - Type I and type II error - Distribution type - Hypothesis test (t-student, means, proportions and ANOVA) - Relationships between variables: correlation, regression.	- Propose questions and hypotheses of interest. - Analyze the data gathered using different statistical tools to answer questions of interest. - Draw conclusions based on the analysis performed.
Readings : [MRo14], [Men14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Men14] Beaver Mendenhall. *Introducción a la probabilidad y estadística*. 13th. Cengage Learning, 2014.

[MRo14] Sheldon M.Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. 5th. Academic Press, 2014.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

FG203. Oratory (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	FG106. Theater. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

En la sociedad competitiva como la nuestra, se exige que la persona sea un comunicador eficaz y sepa utilizar sus potencialidades a fin de resolver problemas y enfrentar los desafíos del mundo moderno dentro de la actividad laboral, intelectual y social. Tener el conocimiento no basta, lo importante es saber comunicarlo y en la medida que la persona sepa emplear sus facultades comunicativas, derivará en éxito o fracaso aquello que tenga que realizar en su desenvolvimiento personal y profesional. Por ello es necesario para lograr un buen decir, recurrir a conocimientos, estrategias y recursos, que debe tener todo orador, para llegar con claridad, precisión y convicción al interlocutor

5. GOALS

- Al término del curso, el alumno será capaz de organizar y asumir la palabra desde la perspectiva del orador, en cualquier situación, en forma más correcta, coherente y adecuada, mediante el uso de conocimientos y habilidades lingüísticas, buscando en todo momento su realización personal y social a través de su expresión, teniendo como base la verdad y la preparación constante.

6. COMPETENCES

f) An ability to communicate effectively. (**Usage**)

n) Apply knowledge of the humanities in their professional work. (**Usage**)

ñ) Understand that the formation of a good professional is not disconnected or opposed but rather contributes to genuine personal growth. This requires the assimilation of solid values, broad spiritual horizons and a deep vision of the cultural environment. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: (3)	
Competences Expected: C24	
Topics	**Learning Outcomes**
- La Oratoria - La función de la palabra. - El proceso de la comunicación. - Bases racionales y emocionales de la oratoria – La expresión oral en la participación. - Fuentes de conocimiento para la oratoria: niveles de cultura general.	- Comprensión: interpretar, ejemplificar y generalizar las bases de la oratoria como fundamento teórico y práctico. [Usage].
Readings : [ME76], [Rod]	

Unit 2: (4)	
Competences Expected: C17	
Topics	**Learning Outcomes**
- Cualidades de un buen orador. - Normas para primeros discursos. - El cuerpo humano como instrumento de comunicación: – La expresión corporal en el discurso – La voz en el discurso. - Oradores con historia y su ejemplo.	- Comprensión: Interpretar, ejemplificar y generalizar conocimientos y habilidades de la comunicación oral mediante la experiencia de grandes oradores y la suya propia. [Usage]. - Aplicación: Implementar, usar, elegir y desempeñar los conocimientos adquiridos para expresarse en público en forma eficiente, inteligente y agradable. [Usage].
Readings : [Rod]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[ME76] A. Monroe and D. Ehninger. *La comunicación oral*. Hispano Europea, 1976.

[Rod] María L. Rodríguez. *Cómo manejar la información en una presentación*.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

ID201. Technical and professional English III (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	-
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	ID102. Technical and professional English II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, undoubtedly English is the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services, English is perhaps the most important practical tool that the student must master from the outset as part of his / her integral education

5. GOALS

- Train the student to understand and hold a conversation.
- Provide techniques of llation of ideas .

6. COMPETENCES

f) An ability to communicate effectively. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Getting to know you! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Present, Past, and Future Times. • Interrogative sentences with Wh-. • Words with more than one meaning. • Parts of the sentence • Expressions for free time	• At the end of the first unit, each of the students, understanding the grammar of present, past and future times, is able to express a greater number of actions in the form of sentences. He is also able to express ideas in the form of questions. Assume the idea of words with more than one meaning. Use social expressions in entertainment situations.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: The way we live! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Simple present tense. • Present Continuous Time. • Collocations. • Vocabulary of the countries of the world. • Expressions of anger. • Connectors.	• Ath the end of the second unit, students having identified the present form of expression recognize the difference between the forms of the same and apply it properly. They describe the countries accurately. They take expressions to show interest. Use connectors to join various ideas.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: It all went wrong! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Past simple tense. • Continuous past tense. • Irregular Verbs. • Time expressions. • Connectors of time.	• At the end of the third unit, students having recognized the characteristics of past times use them properly. They use prefixes and suffixes to create and recognize new words. They describe time in a broad way. They will use conjunctions to unite type ideas.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Let's go shopping! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Expressions of Indefinite Quantity. • Affirmative sentences, Negatives and Questions. • Use of Articles. • Product prices. • Filling of formats and surveys • Expressions for shopping	• At the end of the fourth unit, students having identified the idea of quantity express different situations that involve it. Recognize and apply articles to nouns. They assume the idea of shopping with the help of expressions. They express money prices and ideas. They fill several formats. They express attitudes.
Readings : [SJ02], [Cam06], [Mac99]	

<table>
<tr><td colspan="2"><b>Unit 5: What do you want to do? (0)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C25</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• Verbal Patterns I.
• Future Intentions.
• Verbs of Perception.
• Vocabulary of feelings.
• Expressions of Plans and Ambitions.</td><td>• At the end of the fifth unit, students, from the understanding of the idea of verbal patterns, will elaborate sentences using the necessary elements. They will also assimilate the need to express future intentions. They will acquire vocabulary to describe feelings. Expressions will be presented to describe plans and ambitions.</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Unit 6: The best in the world! (0)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C25</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• What's it like?.
• Adjectives
• Comparative and superlative.
• Synonyms and antonyms.
• Indications of direction .
• Readings.</td><td>• At the end of the sixth unit, students having known the fundamentals of using adjectives, structure sentences with different forms of adjectives in appropriate contexts. They emphasize the difference between types of cities and towns and lifestyles. They use expressions indicating directions.</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Unit 7: Fame! (0)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: C25</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• Present Perfect and Simple Past
• Expressions for, ever, since
• Adverbs
• Expressions that come in pairs
• Short answers
• Celebrities</td><td>• At the end of the seventh unit, students have learned the fundamentals of structuring the present perfect time and differentiate it from the simple past. They emphasize the difference between forms of adjectives. Describe ideas of music. They use expressions to give short answers. They assume the idea of giving extra explanations of the elements of a sentence.</td></tr>
<tr><td colspan="2"><b>Readings :</b> [SJ02], [Cam06], [Mac99]</td></tr>
</table>

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS210. Algorithms and Data Structures (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS113. Computer Science II. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The theoretical foundation of all branches of computing rests on algorithms and data structures, this course will provide participants with an introduction to these topics, thus forming a basis that will serve for the following courses in the career.

5. GOALS

- Make the student understand the importance of algorithms for solving problems.
- Introduce the student to the field of application of data structures.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

a4) (4)

b1) (1)

c1) (1)

7. TOPICS

Unit 1: Graphs (12)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Graph Concept • Directed Graphs and Non-directed Graphs. • Using Graphs. • Measurement of efficiency ,in time and space. • Adjacency matrices. • Tag adjacent matrices. • Adjacency Lists. • Implementation of graphs using adjacency matrices. • Graph Implementation using adjacency lists • Insertion, search and deletion of nodes and edges. • Graph search algorithms.	• Acquire Dexterity to Perform Correct Implementation. [Usage] • Develop knowledge to decide when it is better to use one implementation technique than another. [Usage]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unit 2: Scatter Matrices (8)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Initial concepts. • Dense Matrices • Measurement of Efficiency in Time and Space • Static scatter vs. dynamic matrix creation. • Insert, search, and delete methods.	• Understand the use and implementation of scatter matrices.[Assessment]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

Unit 3: Balanced Trees (16)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• AVL Trees. • Measurement of Efficiency. • Simple and Composite Rotations • Insertion, deletion and search. • Trees B , B+ B* y Patricia.	• Understand the basic functions of these complex structures in order to acquire the capacity for their implementation. [Assessment]
Readings : [Cor+09], [Fag+14], [Knu97], [Knu98]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cor+09] Thomas H. Cormen et al. *Introduction to Algorithms*. Third Edition. ISBN: 978-0-262-53305-8. MIT Press, 2009.

[Fag+14] José Fager et al. *Estructura de datos*. First Edition. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIN), 2014.

[Knu97] Donald E. Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. 3rd. Addison-Wesley Professional, 1997.

[Knu98] Donald E. Knuth. *The art of computer programming, volume 3:Sorting and searching*. 2nd. Addison-Wesley Professional, 1998.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS271. Data Management (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS112. Computer Science I. (2^{nd} Sem) • CS1D2. Discrete Structures II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Information management (IM) plays a major role in almost all areas where computers are used. This area includes the capture, digitization, representation, organization, transformation and presentation of information; Algorithms to improve the efficiency and effectiveness of accessing and updating stored information, data modeling and abstraction, and physical file storage techniques.It also covers information security, privacy, integrity and protection in a shared environment. Students need to be able to develop conceptual and physical data models, determine which (IM) methods and techniques are appropriate for a given problem, and be able to select and implement an appropriate IM solution that reflects all applicable restrictions, including Scalability and usability.

5. GOALS

- That the student learn to represent information in a database prioritizing the efficiency in the recovery of the same.
- That the student learn the fundamental concepts of the management of databases. This includes the design of databases, database languages and the realization of databases.
- Discuss the database model with the base in relational algebra, relational calculus and the study of SQL statements.

6. COMPETENCES

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

b5) (**5**)

d2) (**2**)

d3) (3)

i3) (3)

j2) (2)

j3) (3)

7. TOPICS

Unit 1: Database Systems (14)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Approaches to and evolution of database systems • Components of database systems • Design of core DBMS functions (e.g., query mechanisms, transaction management, buffer management, access methods) • Database architecture and data independence • Use of a declarative query language • Systems supporting structured and/or stream content • Approaches for managing large volumes of data (e.g., noSQL database systems, use of MapReduce).	• Explain the characteristics that distinguish the database approach from the approach of programming with data files [Usage] • Describe the most common designs for core database system components including the query optimizer, query executor, storage manager, access methods, and transaction processor [Usage] • Cite the basic goals, functions, and models of database systems [Usage] • Describe the components of a database system and give examples of their use [Usage] • Identify major DBMS functions and describe their role in a database system [Usage] • Explain the concept of data independence and its importance in a database system [Usage] • Use a declarative query language to elicit information from a database [Usage] • Describe facilities that datatbases provide supporting structures and/or stream (sequence) data, eg, text [Usage] • Describe major approaches to storing and processing large volumes of data [Usage]
Readings : [RC04], [EN04], [RG03], [ER15], [CJ11], [KS02]	

Unit 2: Data Modeling (14)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Data modeling • Conceptual models (e.g., entity-relationship, UML diagrams) • Spreadsheet models • Relational data models • Object-oriented models • Semi-structured data model (expressed using DTD or XML Schema, for example)	• Compare and contrast appropriate data models, including internal structures, for different types of data [Usage] • Describe concepts in modeling notation (eg, Entity-Relation Diagrams or UML) and how they would be used [Usage] • Define the fundamental terminology used in the relational data model [Usage] • Describe the basic principles of the relational data model [Usage] • Apply the modeling concepts and notation of the relational data model [Usage] • Describe the main concepts of the OO model such as object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning [Usage] • Describe the differences between relational and semi-structured data models [Usage] • Give a semi-structured equivalent (eg, in DTD or XML Schema) for a given relational schema [Usage]
Readings : [SW04], [EN04], [KS02]	

Unit 3: Indexing (4)	
Competences Expected: b,d,i	
Topics	**Learning Outcomes**
• The impact of indices on query performance • The basic structure of an index • Keeping a buffer of data in memory • Creating indexes with SQL • Indexing text • Indexing the web (e.g., web crawling)	• Generate an index file for a collection of resources [Usage] • Explain the role of an inverted index in locating a document in a collection [Usage] • Explain how stemming and stop words affect indexing [Usage] • Identify appropriate indices for given relational schema and query set [Usage] • Estimate time to retrieve information, when indices are used compared to when they are not used [Usage] • Describe key challenges in web crawling, eg, detecting duplicate documents, determining the crawling frontier [Usage]
Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unit 4: Relational Databases (14)	
Competences Expected: b,d,i	
Topics	**Learning Outcomes**
• Mapping conceptual schema to a relational schema • Entity and referential integrity • Relational algebra and relational calculus • Relational Database design • Functional dependency • Decomposition of a schema; lossless-join and dependency-preservation properties of a decomposition • Candidate keys, superkeys, and closure of a set of attributes • Normal forms (BCNF) • Multi-valued dependency (4NF) • Join dependency (PJNF, 5NF) • Representation theory	• Prepare a relational schema from a conceptual model developed using the entity- relationship model [Usage] • Explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint (including definition of the concept of a foreign key) [Usage] • Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division) [Usage] • Write queries in the relational algebra [Usage] • Write queries in the tuple relational calculus [Usage] • Determine the functional dependency between two or more attributes that are a subset of a relation [Usage] • Connect constraints expressed as primary key and foreign key, with functional dependencies [Usage] • Compute the closure of a set of attributes under given functional dependencies [Usage] • Determine whether a set of attributes form a superkey and/or candidate key for a relation with given functional dependencies [Usage] • Evaluate a proposed decomposition, to say whether it has lossless-join and dependency-preservation [Usage] • Describe the properties of BCNF, PJNF, 5NF [Usage] • Explain the impact of normalization on the efficiency of database operations especially query optimization [Usage] • Describe what is a multi-valued dependency and what type of constraints it specifies [Usage]
Readings : [WM01], [RG03], [ER15], [CJ11], [KS02]	

Unit 5: Query Languages (12)	
Competences Expected: b,d,i,j	
Topics	**Learning Outcomes**
• Overview of database languages • SQL (data definition, query formulation, update sub-language, constraints, integrity) • Selections • Projections • Select-project-join • Aggregates and group-by • Subqueries • QBE and 4th-generation environments • Different ways to invoke non-procedural queries in conventional languages • Introduction to other major query languages (e.g., XPATH, SPARQL) • Stored procedures	• Create a relational database schema in SQL that incorporates key, entity integrity, and referential integrity constraints [Usage] • Use SQL to create tables and retrieve (SELECT) information from a database [Usage] • Evaluate a set of query processing strategies and select the optimal strategy [Usage] • Create a non-procedural query by filling in templates of relations to construct an example of the desired query result [Usage] • Embed object-oriented queries into a stand-alone language such as C++ or Java (eg, SELECT Col-Method() FROM Object) [Usage] • Write a stored procedure that deals with parameters and has some control flow, to provide a given functionality [Usage]
Readings : [Die01], [EN04], [Cel05], [KS02]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cel05] Joe Celko. *Joe Celko's SQL Programming Style*. Elsevier, 2005.

[CJ11] Date C.J. *SQL and Relational Theory: How to Write Accurate SQL Code*. O'Reilly Media, 2011.

[Die01] Suzanne W Dietrich. *Understanding Relational Database Query Languages, First Edition*. Prentice Hall, 2001.

[EN04] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems, Fourth Edition*. Addison Wesley, 2004.

[ER15] Jim Webber Emil Eifrem and Ian Robinson. *Graph Databases*. 2nd. O'Reilly Media, 2015.

[KS02] Henry F. Korth and Abraham Silberschatz. *Fundamentos de Base de Datos*. McGraw-Hill, 2002.

[RC04] Peter Rob and Carlos Coronel. *Database Systems: Design, Implementation and Management, Sixth Edition.* Morgan Kaufmann, 2004.

[RG03] Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems.* 3rd. McGraw-Hill, 2003.

[SW04] Graeme Simsion and Graham Witt. *Data Modeling Essentials, Third Edition.* Morgan Kaufmann, 2004.

[WM01] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases, Second Edition.* Springer, 2001.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS2S1. Operating systems (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS221. Computer Systems Architecture. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

An Operating System (OS) manages the computing resources to complete the execution of multiple applications and their associated processes. This course teaches the design of modern operating systems; and introduces their fundamental concepts covering multiple-program execution, scheduling, memory management, file systems, and security. Also, the course includes programming activities on a minimal operating system to solve problems and extend its functionality. Notice that these activities require much time to complete. However, working on them provides valuable insight into operating systems.

5. GOALS

- Study the design of modern operating systems.

- Provide a practical experience by designing and implementing a minimal operating system.

6. COMPETENCES

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Overview of Operating Systems (3)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Role and purpose of the operating system • Functionality of a typical operating system • Mechanisms to support client-server models. • Design issues (efficiency, robustness, flexibility, portability, security, compatibility) • Influences of security, networking, multimedia, windowing systems	• Explain the objectives and functions of modern operating systems [Familiarity] • Analyze the tradeoffs inherent in operating system design [Assessment] • Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve [Familiarity] • Discuss networked, client-server, distributed operating systems and how they differ from single user operating systems [Familiarity] • Identify potential threats to operating systems and the security features design to guard against them [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 2: Operating System Principles (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Operating Sistems Structure (monolithic, layered, modular, micro-kernel models) • Abstractions, processes, and resources • Concepts of application program interfaces (APIs) • The evolution of hardware/software techniques and application needs • Device organization • Interrupts: methods and implementations • Concept of user/system state and protection, transition to kernel mode	• Explain the concept of a logical layer [Familiarity] • Explain the benefits of building abstract layers in hierarchical fashion [Familiarity] • Describe the value of APIs and middleware [Familiarity] • Describe how computing resources are used by application software and managed by system software [Familiarity] • Contrast kernel and user mode in an operating system [Assessment] • Discuss the advantages and disadvantages of using interrupt processing [Familiarity] • Explain the use of a device list and driver I/O queue [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 3: Concurrency (9)	
Competences Expected: b	
Topics	**Learning Outcomes**
• States diagrams • Structures (ready list, process control blocks, and so forth) • Dispatching and context switching • The role of interrupts • Managing atomic access to OS objects • Implementing synchronization primitives • Multiprocessor issues (spin-locks, reentrancy)	• Describe the need for concurrency within the framework of an operating system [Familiarity] • Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks [Usage] • Summarize the range of mechanisms that can be employed at the operating system level to realize concurrent systems and describe the benefits of each [Familiarity] • Explain the different states that a task may pass through and the data structures needed to support the management of many tasks [Familiarity] • Summarize techniques for achieving synchronization in an operating system (eg, describe how to implement a semaphore using OS primitives) [Familiarity] • Describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system [Familiarity] • Create state and transition diagrams for simple problem domains [Usage]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 4: Scheduling and Dispatch (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Preemptive and non-preemptive scheduling • Schedulers and policies • Processes and threads • Deadlines and real-time issues	• Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes [Assessment] • Describe relationships between scheduling algorithms and application domains [Familiarity] • Discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O [Familiarity] • Describe the difference between processes and threads [Familiarity] • Compare and contrast static and dynamic approaches to real-time scheduling [Assessment] • Discuss the need for preemption and deadline scheduling [Familiarity] • Identify ways that the logic embodied in scheduling algorithms are applicable to other domains, such as disk I/O, network scheduling, project scheduling, and problems beyond computing [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 5: Memory Management (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Review of physical memory and memory management hardware • Working sets and thrashing • Caching	• Explain memory hierarchy and cost-performance trade-offs [Familiarity] • Summarize the principles of virtual memory as applied to caching and paging [Familiarity] • Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed [Assessment] • Defend the different ways of allocating memory to tasks, citing the relative merits of each [Familiarity] • Describe the reason for and use of cache memory (performance and proximity, different dimension of how caches complicate isolation and VM abstraction) [Familiarity] • Discuss the concept of thrashing, both in terms of the reasons it occurs and the techniques used to recognize and manage the problem [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 6: Security and Protection (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Overview of system security • Policy/mechanism separation • Security methods and devices • Protection, access control, and authentication • Backups	• Articulate the need for protection and security in an OS [Familiarity] • Summarize the features and limitations of an operating system used to provide protection and security [Familiarity] • Explain the mechanisms available in an OS to control access to resources (cross reference IAS/Security Architecture and Systems Administration/Access Control/Configuring systems to operate securely as an IT system) [Familiarity] • Carry out simple system administration tasks according to a security policy, for example creating accounts, setting permissions, applying patches, and arranging for regular backups (cross reference IAS/Security Architecture and Systems Administration) [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 7: Virtual Machines (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Types of virtualization (including Hardware/Software, OS, Server, Service, Network) • Paging and virtual memory • Virtual file systems • Hypervisors • Portable virtualization; emulation vs. isolation • Cost of virtualization	• Explain the concept of virtual memory and how it is realized in hardware and software [Familiarity] • Differentiate emulation and isolation [Familiarity] • Evaluate virtualization trade-offs [Assessment] • Discuss hypervisors and the need for them in conjunction with different types of hypervisors [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 8: Device Management (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Characteristics of serial and parallel devices • Abstracting device differences • Buffering strategies • Direct memory access • Recovery from failures	• Explain the key difference between serial and parallel devices and identify the conditions in which each is appropriate [Familiarity] • Identify the relationship between the physical hardware and the virtual devices maintained by the operating system [Familiarity] • Explain buffering and describe strategies for implementing it [Familiarity] • Differentiate the mechanisms used in interfacing a range of devices (including hand-held devices, networks, multimedia) to a computer and explain the implications of these for the design of an operating system [Familiarity] • Describe the advantages and disadvantages of direct memory access and discuss the circumstances in which its use is warranted [Familiarity] • Identify the requirements for failure recovery [Familiarity] • Implement a simple device driver for a range of possible devices [Usage]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 9: File Systems (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Files: data, metadata, operations, organization, buffering, sequential, nonsequential. • Directories: contents and structure. • File systems: partitioning, mount/unmount, virtual file systems. • Standard implementation techniques • Memory-mapped files • Special-purpose file systems. • Naming, searching, access, backups • Journaling and log-structured file systems	• Describe the choices to be made in designing file systems [Familiarity] • Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each [Assessment] • Summarize how hardware developments have led to changes in the priorities for the design and the management of file systems [Familiarity] • Summarize the use of journaling and how log-structured file systems enhance fault tolerance [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 10: Real Time and Embedded Systems (6)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Process and task scheduling • Memory/disk management requirements in a real-time environment • Failures, risks, and recovery. • Special concerns in real-time systems	• Describe what makes a system a real-time system [Familiarity] • Explain the presence of and describe the characteristics of latency in real-time systems [Familiarity] • Summarize special concerns that real-time systems present, including risk, and how these concerns are addressed [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 11: Fault Tolerance (3)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Fundamental concepts: reliable and available systems • Spatial and temporal redundancy • Methods used to implement fault tolerance • Examples of OS mechanisms for detection, recovery, restart to implement fault tolerance, use of these techniques for the OS's own services.	• Explain the relevance of the terms fault tolerance, reliability, and availability [Familiarity] • Outline the range of methods for implementing fault tolerance in an operating system [Familiarity] • Explain how an operating system can continue functioning after a fault occurs [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

Unit 12: System Performance Evaluation (3)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Why system performance needs to be evaluated? • What is to be evaluated? • Systems performance policies, e.g., caching, paging, scheduling, memory management, and security • Evaluation models: deterministic, analytic, simulation, or implementation-specific • How to collect evaluation data (profiling and tracing mechanisms)	• Describe the performance measurements used to determine how a system performs [Familiarity] • Explain the main evaluation models used to evaluate a system [Familiarity]
Readings : [Avi12], [Sta05], [Tan06], [Tan01], [AD14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[AD14] Thomas Anderson and Michael Dahlin. *Operating Systems: Principles and Practice.* 2nd. Recursive Books, 2014. ISBN: 978-0985673529.

[Avi12] Greg Gagne Avi Silberschatz Peter Baer Galvin. *Operating System Concepts, 9/E.* John Wiley & Sons, Inc., 2012. ISBN: 978-1-118-06333-0.

[Sta05] William Stallings. *Operating Systems: Internals and Design Principles, 5/E.* Prentice Hall, 2005. ISBN: 0-13-147954-7.

[Tan01] Andrew S. Tanenbaum. *Modern Operating Systems, 4/E.* Prentice Hall, 2001. ISBN: 0-13-031358-0.

[Tan06] Andrew S. Tanenbaum. *Operating Systems Design and Implementation, 3/E.* Prentice Hall, 2006. ISBN: 0-13-142938-8.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

FG350. Leadership and Performance (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	2
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	FG203. Oratory. (3rd Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

En la actualidad las diferentes organizaciones en el mundo exigen a sus integrantes el ejercicio de liderazgo, esto significa asumir los retos asignados con eficacia y afán de servicio, siendo estas exigencias necesarias para la búsqueda de una sociedad más justa y reconciliada. Este desafío, pasa por la necesidad de formar a nuestros alumnos con un recto conocimiento de sí mismos, con capacidad de juzgar objetivamente la realidad y de proponer orientaciones que busquen modificar positivamente el entorno.

5. GOALS

- Desarrollar conocimientos, criterios, capacidades y actitudes para ejercer liderazgo, con el objeto de lograr la eficacia y servicio en los retos asignados, contribuyendo así en la construcción de una mejor sociedad.

6. COMPETENCES

d) An ability to function on multidisciplinary teams. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

ñ) Understand that the formation of a good professional is not disconnected or opposed but rather contributes to genuine personal growth. This requires the assimilation of solid values, broad spiritual horizons and a deep vision of the cultural environment. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: (15)	
Competences Expected: C18,C24	
Topics	**Learning Outcomes**
• Teorías de Liderazgo: • Definición de Liderazgo. • Fundamentos de Liderazgo. • Visión integral del Ser Humano y Motivos de la acción. • La práctica de la Virtud en el ejercicio de Liderazgo.	• Analizar y comprender las bases teóricas del ejercicio de Liderazgo.[Familiarity] • En base a lo comprendido, asumir la actitud correcta para llevarlo a la práctica.[Familiarity] • Iniciar un proceso de autoconocimiento orientado a descubrir rasgos de liderazgo en sí mismo.[Familiarity]
Readings : [Pil02], [Man09], [Ale09], [D S], [Alf10]	

Unit 2: (15)	
Competences Expected: C17,C18,C24	
Topics	**Learning Outcomes**
• Teoría de las Competencias • Reconocimiento de Competencias • Plan de Desarrollo • Modelos Mentales • Necesidades Emocionales • Perfiles Emocionales • Vicios Motivacionales	• Conocer y Desarrollar competencias de Liderazgo, centradas en lograr la eficacia, sin dejar de lado el deber de servicio con los demás.[Familiarity] • Reconocer las tendencias personales y grupales necesarias para el ejercicio de Liderazgo.[Familiarity]
Readings : [Wil09], [Lui08], [Pil02], [Mar07]	

Unit 3: (18)	
Competences Expected: C18,C24	
Topics	**Learning Outcomes**
• La relación personal con el equipo • Liderazgo integral • Acompañamiento y discipulado • Fundamentos de unidad	• Desarrollar habilidades para el trabajo en equipo[Familiarity]
Readings : [Gol12], [**CardonaP**], [**Hersey**], [Hun10], [Haw12], [**Ginebra**]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Ale09] Dianine-Havard Alexandre. *Perfil del Líder. Hacia un Liderazgo Virtuoso.* Ediciones Urano S.A, 2009.

[Alf10] Sonnenfeld Alfred. *Liderazgo Ético. La Sabiduría de decidir bien.* Ediciones Encuentro S.A Madrid y Nueva Revista de Madrid, 2010.

[D S] SJ Anthony. D' Souza. *Descubre tu Liderazgo.* Editorial Sal Terrae.

[Gol12] D. Goleman. *Inteligencia emocional.* Editorial Kairós., 2012.

[Haw12] Peter. Hawkins. *Coaching y liderazgo de equipos: coaching para un liderazgo con capacidad de transformación.* Ediciones Granica, 2012.

[Hun10] Phil. Hunsaker. *El nuevo arte de gestionar equipos: Un enfoque actual para guiar y motivar con éxito.* 2010.

[Lui08] Huete Luis. *Construye tu Sueño.* LID Editorial Empresarial, 2008.

[Man09] Ferreiro Pablo/Alcázar Manuel. *Gobierno de Personas en la Empresa.* Ediciones Universidad de Navarra EUNSA, 2009.

[Mar07] Chinchilla Nuria/Moragas Maruja. *Dueños de Nuestro Destino.* Editorial Ariel, 2007.

[Pil02] Cardona Pablo/García Lombardi Pilar. *Cómo desarrollar las Competencias de Liderazgo.* PAD Lima- Perú, Tercera Edición., 2002.

[Wil09] Cardona Pablo/ Helen Wilkinson. *Creciendo como Líder.* Ediciones Universidad de Navarra S.A (EUNSA), Primera Edición, 2009.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

ID202. English IV (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	-
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	ID201. Technical and professional English III. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, English is undoubtedly the most practical because it is spoken around all the world. There is no country where it is not spoken. In addition to being vital to your professional career

5. GOALS

- Increase the level of conversation in different subjects, in the students. As well as the ability to write and read documentation of all kinds.
- Bring the student to a more intense expression in the language domain.

6. COMPETENCES

f) An ability to communicate effectively. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Do and don't! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Mode Auxiliaries should, must and have got to. • Affirmative, negative and interrogative sentences with modals. • Terms for formal letters. • Parts of short answers. • Expressions for occupations.	• At the end of the eighth unit, each of the students, understanding the grammar of should and must auxiliaries, is able to express a greater number of actions in an obligatory and suggestive way. Also be able to express ideas describing occupations. Assumes the need to write formal letters
Readings : [SJ02], [Cam06], [Mac99]	

Unit 2: Going places! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Present and Future Present Time with Will • First conditional • Collocations • Vocabulary of prepositions of place and time • Expressions of connection of ideas	• At the end of the ninth unit, students having identified how to express present recognize the difference between future forms and apply them properly. They describe conditions accurately. They assume expressions to show place location. They use expressions of time and connectors to unite several ideas.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 3: Scared to death! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Infinitive and gerund verb patterns • What + Infinitive • Something + infinitive • Expressions of feelings • Exclamations of surprise	• At the end of the tenth unit of students, the chapters recognize and use the patterns of times in the past properly. They use exclamation marks. And describe feelings. They will use conjunctions to unite type ideas.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 4: Things that changed the world! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Passive Voice Â Â Â Â Â Â • Affirmative Prayers, Negatives and Questions Â Â Â Â Â Â • Use of participles, verbs and nouns that go together Â Â Â Â Â Â • Signals. Signs and notes Â Â Â Â Â Â • Summaries Â Â Â Â Â Â • Expressions to indicate prohibition	• At the end of the eleventh unit ,the students having identified the idea of passive actions describe actions appropriately in diverse situations that involve it. They recognize and apply participations. They assume the idea of respecting public signs and signals. They express ideas of habits. They make summaries.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 5: Dreams and reality! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Second Conditional Â Â Â Â Â Â • Auxiliar of mode ”might” Â Â Â Â Â Â • Phrase Verbs Â Â Â Â Â Â • Social expressions vocabulary Â Â Â Â Â Â • Adverbs Â Â Â Â Â Â • Expressions to give advice	• At the end of the twelfth unit, students, starting from understanding the idea of Conditionals and expressing the possibility of elaborating sentences using the necessary elements. They will also assimilate the need for verbal phrases (2 word verbs). They will acquire vocabulary to describe social expressions.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 6: Making a living! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Present Perfect Continuous Â Â Â Â Â Â • Present Continuous Â Â Â Â Â Â • Occupations Â Â Â Â Â Â • Word formation Â Â Â Â Â Â • Adverbs Â Â Â Â Â Â • Expressions of use on the phone	• At the end of the thirteenth unit, they structure sentences with actions that include present and past in appropriate contexts. They emphasize the difference between types of occupations. Use appropriate expressions for telephone conversations.
Readings : [SJ02], [Cam06], [Mac99]	

Unit 7: All you need is love! (0)	
Competences Expected: C25	
Topics	**Learning Outcomes**
• Past Perfect and Past Simple Â Â Â Â Â Â • Report Expressions Â Â Â Â Â Â • Expressions of words in different contexts Â Â Â Â Â Â • Short and formal farewells Â Â Â Â Â Â • Love Stories	• At the end of the fourteenth unit, students having learned the fundamentals of structuring past perfect time, differentiate it from the simple past. They emphasize the difference between words in different contexts. Describe farewell ideas. They use expressions to write love stories. They assume the idea of giving and doing interviews.
Readings : [SJ02], [Cam06], [Mac99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02] Liz Soars and John. *American Headway N 2 Student Book*. Editorial Oxford, 2002.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS212. Analysis and Design of Algorithms (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS210. Algorithms and Data Structures. (4^{th} Sem) • MA101. Math II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

An algorithm is, essentially, a well-defined set of rules or instructions that allow solving a computational problem. The theoretical study of the performance of the algorithms and the resources used by them, usually time and space, allows us to evaluate if an algorithm is suitable for solving a specific problem, comparing it with other algorithms for the same problem or even delimiting the boundary between Viable and impossible. This matter is so important that even Donald E. Knuth defined Computer Science as the study of algorithms. This course will present the most common techniques used in the analysis and design of efficient algorithms, with the purpose of learning the fundamental principles of the design, implementation and analysis of algorithms for the solution of computational problems

5. GOALS

- Develop the ability to evaluate the complexity and quality of algorithms proposed for a given problem.
- Study the most representative, introductory algorithms of the most important classes of problems treated in computation.
- Develop the ability to solve algorithmic problems using the fundamental principles of algorithm design learneds.
- Be able to answer the following questions when a new algorithm is presented: How good is the performance ?, Is there a better way to solve the problem?

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

a8) (**8**)

a9) (**9**)

b1) (**1**)

b3) (**3**)

b4) (**4**)

7. TOPICS

Unit 1: Basic Analysis (10)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Differences among best, expected, and worst case behaviors of an algorithm • Asymptotic analysis of upper and expected complexity bounds • Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential • Asymptotic Notation • Analysis of iterative and recursive algorithms • Inductive proofs and correctness of algorithms • Master Theorem and Recursion Trees	• Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm [Assessment] • Determine informally the time and space complexity of different algorithms [Assessment] • List and contrast standard complexity classes [Assessment] • Explain the use of big omega, big theta, and little o notation to describe the amount of work done by an algorithm [Assessment] • Analyze worst-case running times of algorithms using asymptotic analysis [Assessment] • Use recurrence relations to determine the time complexity of recursively defined algorithms [Assessment] • Solve elementary recurrence relations, eg, using some form of a Master Theorem [Assessment] • Argue the correctness of algorithms using inductive proofs [Assessment]
Readings : [KT05], [DPV06], [RS09], [SF13], [Knu97]	

Unit 2: Algorithmic Strategies (30)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Brute-force algorithms • Greedy algorithms • Divide-and-conquer • Dynamic Programming	• For each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply [Assessment] • Use a greedy approach to solve an appropriate problem and determine if the greedy rule chosen leads to an optimal solution [Assessment] • Use a divide-and-conquer algorithm to solve an appropriate problem [Assessment] • Use dynamic programming to solve an appropriate problem [Assessment] • Determine an appropriate algorithmic approach to a problem [Assessment]
Readings : [KT05], [DPV06], [RS09], [Als99]	

Unit 3: Fundamental Data Structures and Algorithms (6)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Graphs and graph algorithms – Maximum and minimum cut problem – Local search • Cache oblivious algorithms • Number theory and cryptography	• Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data [Familiarity] • Solve problems using fundamental graph algorithms, including depth-first and breadth-first search [Assessment] • Demonstrate the ability to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in a particular context [Assessment] • Solve problems using graph algorithms, including single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm [Assessment]
Readings : [KT05], [DPV06], [RS09], [SW11], [GT09]	

Unit 4: Basic Automata Computability and Complexity (2)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Introduction to the P and NP classes and the P vs. NP problem • Introduction to the NP-complete class and exemplary NP-complete problems (e.g., SAT, Knapsack) • Reductions	• Define the classes P and NP [Familiarity] • Explain the significance of NP-completeness [Familiarity]
Readings : [KT05], [DPV06], [RS09]	

Unit 5: Advanced Data Structures Algorithms and Analysis (12)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Graphs (e.g, topological sort, finding strongly connected components, matching) • Randomized algorithms • Amortized analysis • Probabilistic analysis • Approximation Algorithms • Linear Programming	• Understand the mapping of real-world problems to algorithmic solutions (eg, as graph problems, linear programs, etc) [Familiarity] • Select and apply advanced analysis techniques (eg, amortized, probabilistic, etc) to algorithms [Usage]
Readings : [KT05], [DPV06], [RS09], [Tar83], [Raw92]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Als99] H. Alsuwaiyel. *Algorithms: Design Techniques and Analysis*. World Scientific, 1999. ISBN: 9789810237400.

[DPV06] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Education, 2006. ISBN: 9780073523408.

[GT09] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples*. 2nd. John Wiley & Sons, Inc., 2009. ISBN: 0470088540, 9780470088548.

[Knu97] D.E. Knuth. *The Art of Computer Programming: Fundamental algorithms Vol 1*. Third Edition. Addison-Wesley, 1997. ISBN: 9780201896831. URL: `http://www-cs-faculty.stanford/~knuth/taocp.html`.

[KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN: 0321295358.

[Raw92] G.J.E. Rawlins. *Compared to What?: An Introduction to the Analysis of Algorithms*. Computer Science Press, 1992. ISBN: 9780716782438.

[RS09] Thomas H. Cormen; Charles E. Leiserson ; Ronald L. Rivest and Clifford Stein. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844.

[SF13] R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms*. Pearson Education, 2013. ISBN: 9780133373486.

[SW11] R. Sedgewick and K. Wayne. *Algorithms*. Pearson Education, 2011. ISBN: 9780132762564.

[Tar83] Robert Endre Tarjan. *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics, 1983. ISBN: 0-89871-187-8.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS231. Networking and Communication (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS2S1. Operating systems . (4th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The ever-growing development of communication and information technologies means that there is a marked tendency to establish more computer networks that allow better information management..

In this second course, participants will be introduced to the problems of communication between computers, through the study and implementation of communication protocols such as TCP / IP and the implementation of software on these protocols

5. GOALS

- That the student implements and / or modifies a data communication protocols.
- That the student master the data transmission techniques used by the existing network protocols.
- That the student knows the latest trends in networks that are being applied on the Internet.

6. COMPETENCES

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

<table>
<tr><th colspan="2">Unit 1: Introduction (5)</th></tr>
<tr><th colspan="2">Competences Expected: b,c</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>• Organization of the Internet (Internet Service Providers, Content Providers, etc.)
• Switching techniques (e.g., circuit, packet)
• Physical pieces of a network, including hosts, routers, switches, ISPs, wireless, LAN, access point, and firewalls
• Layering principles (encapsulation, multiplexing)
• Roles of the different layers (application, transport, network, datalink, physical)</td><td>• Articulate the organization of the Internet [Familiarity]
• List and define the appropriate network terminology [Familiarity]
• Describe the layered structure of a typical networked architecture [Familiarity]
• Identify the different types of complexity in a network (edges, core, etc) [Familiarity]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [KR13]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 2: Networked Applications (5)</th></tr>
<tr><th colspan="2">Competences Expected: b,c,i</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>• Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.)
• Distributed applications (client/server, peer-to-peer, cloud, etc.)
• HTTP as an application layer protocol
• Multiplexing with TCP and UDP
• Socket APIs</td><td>• List the differences and the relations between names and addresses in a network [Familiarity]
• Define the principles behind naming schemes and resource location [Familiarity]
• Implement a simple client-server socket-based application [Usage]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [KR13]</td></tr>
</table>

<table>
<tr><th colspan="2">Unit 3: Reliable Data Delivery (10)</th></tr>
<tr><th colspan="2">Competences Expected: C6,b,c,i</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>• Error control (retransmission techniques, timers)
• Flow control (acknowledgements, sliding window)
• Performance issues (pipelining)
• TCP</td><td>• Describe the operation of reliable delivery protocols [Familiarity]
• List the factors that affect the performance of reliable delivery protocols [Familiarity]
• Design and implement a simple reliable protocol [Usage]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [KR13]</td></tr>
</table>

Unit 4: Routing and Forwarding (12)

Competences Expected: b,c,i

Topics	Learning Outcomes
• Routing versus forwarding • Static routing • Internet Protocol (IP) • Scalability issues (hierarchical addressing)	• Describe the organization of the network layer [Familiarity] • Describe how packets are forwarded in an IP network [Familiarity] • List the scalability benefits of hierarchical addressing [Familiarity]

Readings : [KR13]

Unit 5: Local Area Networks (10)

Competences Expected: b,c

Topics	Learning Outcomes
• Multiple Access Problem • Common approaches to multiple access (exponential-backoff, time division multiplexing, etc) • Local Area Networks • Ethernet • Switching	• Describe how frames are forwarded in an Ethernet network [Familiarity] • Describe the interrelations between IP and Ethernet [Familiarity] • Describe the steps used in one common approach to the multiple access problem [Familiarity]

Readings : [KR13]

Unit 6: Resource Allocation (12)

Competences Expected: b,c,i

Topics	Learning Outcomes
• Need for resource allocation • Fixed allocation (TDM, FDM, WDM) versus dynamic allocation • End-to-end versus network assisted approaches • Fairness • Principles of congestion control • Approaches to Congestion (e.g., Content Distribution Networks)	• Describe how resources can be allocated in a network [Familiarity] • Describe the congestion problem in a large network [Familiarity] • Compare and contrast fixed and dynamic allocation techniques [Familiarity] • Compare and contrast current approaches to congestion [Familiarity]

Readings : [KR13]

Unit 7: Mobility (5)	
Competences Expected: b,c	
Topics	**Learning Outcomes**
• Principles of cellular networks • 802.11 networks • Issues in supporting mobile nodes (home agents)	• Describe the organization of a wireless network [Familiarity] • Describe how wireless networks support mobile users [Familiarity]
Readings : [KR13], [Cha16]	

Unit 8: Social Networking (5)	
Competences Expected: b,c,i	
Topics	**Learning Outcomes**
• Social networks overview • Example social network platforms • Structure of social network graphs • Social network analysis	• Discuss the key principles (such as membership, trust) of social networking [Familiarity] • Describe how existing social networks operate [Familiarity] • Construct a social network graph from network data [Usage] • Analyze a social network to determine who the key people are [Usage] • Evaluate a given interpretation of a social network question with associated data [Familiarity]
Readings : [KR13], [Kad11]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cha16] Paresh Chayapathi Rajendra; Syed F. Hassan; Shah. *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional; 1 edition, 2016. ISBN: 978-0134463056.

[Kad11] Charles Kadushin. *Understanding Social Networks: Theories, Concepts, And Findings*. Oxford University Press, Usa; 1 edition, 2011. ISBN: 978-0195379471.

[KR13] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-down Approach*. 7th. Always learning. Pearson, 2013. ISBN: 978-0133594140.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS261. Intelligent Systems (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA203. Statistics and Probabilities. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Research in Artificial Intelligence has led to the development of numerous relevant tonic, aimed at the automation of human intelligence, giving a panoramic view of different algorithms that simulate the different aspects of the behavior and the intelligence of the human being.

5. GOALS

- Evaluate the possibilities of simulation of intelligence, for which the techniques of knowledge modeling will be studied.

- Build a notion of intelligence that later supports the tasks of your simulation.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Familiarity**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Familiarity**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Fundamental Issues (2)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Overview of AI problems, examples of successful recent AI applications • What is intelligent behavior? – The Turing test – Rational versus non-rational reasoning • Problem characteristics – Fully versus partially observable – Single versus multi-agent – Deterministic versus stochastic – Static versus dynamic – Discrete versus continuous • Nature of agents – Autonomous versus semi-autonomous – Reflexive, goal-based, and utility-based – The importance of perception and environmental interactions • Philosophical and ethical issues.	• Describe Turing test and the “Chinese Room” thought experiment [Usage] • Determing the characteristics of a given problem that an intelligent systems must solve [Usage]
Readings : [De 06], [Pon+14]	

Unit 2: Agents (2)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Definitions of agents • Agent architectures (e.g., reactive, layered, cognitive) • Agent theory • Rationality, game theory – Decision-theoretic agents – Markov decision processes (MDP) • Software agents, personal assistants, and information access – Collaborative agents – Information-gathering agents – Believable agents (synthetic characters, modeling emotions in agents) • Learning agents • Multi-agent systems – Collaborating agents – Agent teams – Competitive agents (e.g., auctions, voting) – Swarm systems and biologically inspired models	• List the defining characteristics of an intelligent agent [Usage] • Characterize and contrast the standard agent architectures [Usage] • Describe the applications of agent theory to domains such as software agents, personal assistants, and believable agents [Usage] • Describe the primary paradigms used by learning agents [Usage] • Demonstrate using appropriate examples how multi-agent systems support agent interaction [Usage]
Readings : [Nil01], [RN03], [Pon+14]	

Unit 3: Basic Search Strategies (2)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Problem spaces (states, goals and operators), problem solving by search • Factored representation (factoring state into variables) • Uninformed search (breadth-first, depth-first, depth-first with iterative deepening) • Heuristics and informed search (hill-climbing, generic best-first, A*) • Space and time efficiency of search • Two-player games (introduction to minimax search) • Constraint satisfaction (backtracking and local search methods)	• Formulate an efficient problem space for a problem expressed in natural language (eg, English) in terms of initial and goal states, and operators [Usage] • Describe the role of heuristics and describe the trade-offs among completeness, optimality, time complexity, and space complexity [Usage] • Describe the problem of combinatorial explosion of search space and its consequences [Usage] • Compare and contrast basic search issues with game playing issues [Usage]
Readings : [Nil01], [Pon+14]	

<table>
<tr><td colspan="2"><b>Unit 4: Advanced Search (18)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: a,j</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• Stochastic search
– Simulated annealing
– Genetic algorithms
– Monte-Carlo tree search
• Constructing search trees, dynamic search space, combinatorial explosion of search space
• Implementation of A* search, beam search
• Minimax search, alpha-beta pruning
• Expectimax search (MDP-solving) and chance nodes</td><td>• Design and implement a genetic algorithm solution to a problem [Usage]
• Design and implement a simulated annealing schedule to avoid local minima in a problem [Usage]
• Design and implement A*,beam search to solve a problem [Usage]
• Apply minimax search with alpha-beta pruning to prune search space in a two-player game [Usage]
• Compare and contrast genetic algorithms with classic search techniques [Usage]
• Compare and contrast various heuristic searches vis-a-vis applicability to a given problem [Usage]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [Gol89], [Nil01], [RN03], [Pon+14]</td></tr>
</table>

<table>
<tr><td colspan="2"><b>Unit 5: Reasoning Under Uncertainty (18)</b></td></tr>
<tr><td colspan="2"><b>Competences Expected: a,j</b></td></tr>
<tr><td><b>Topics</b></td><td><b>Learning Outcomes</b></td></tr>
<tr><td>• Review of basic probability
• Random variables and probability distributions
– Axioms of probability
– Probabilistic inference
– Bayes' Rule
• Conditional Independence
• Knowledge representations
– Bayesian Networks
∗ Exact inference and its complexity
∗ Randomized sampling (Monte Carlo) methods (e.g. Gibbs sampling)
– Markov Networks
– Relational probability models
– Hidden Markov Models</td><td>• Apply Bayes' rule to determine the probability of a hypothesis given evidence [Usage]
• Explain how conditional independence assertions allow for greater efficiency of probabilistic systems [Usage]
• Identify examples of knowledge representations for reasoning under uncertainty [Usage]
• State the complexity of exact inference Identify methods for approximate inference [Usage]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [KF09], [RN03]</td></tr>
</table>

Unit 6: Basic Machine Learning (4)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Definition and examples of broad variety of machine learning tasks, including classification • Inductive learning • Simple statistical-based learning, such as Naive Bayesian Classifier, decision trees • The over-fitting problem • Measuring classifier accuracy	• List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised [Usage] • Identify examples of classification tasks, including the available input features and output to be predicted [Usage] • Explain the difference between inductive and deductive learning [Usage] • Describe over-fitting in the context of a problem [Usage] • Apply the simple statistical learning algorithm such as Naive Bayesian Classifier to a classification task and measure the classifier's accuracy [Usage]
Readings : [Mit98], [RN03], [Pon+14]	

Unit 7: Advanced Machine Learning (20)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Definition and examples of broad variety of machine learning tasks • General statistical-based learning, parameter estimation (maximum likelihood) • Inductive logic programming (ILP) • Supervised learning – Learning decision trees – Learning neural networks – Support vector machines (SVMs) • Unsupervised Learning and clustering – EM – K-means – Self-organizing maps • Semi-supervised learning • Learning graphical models • Performance evaluation (such as cross-validation, area under ROC curve) • Application of Machine Learning algorithms to Data Mining (cross-reference IM/Data Mining)	• Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised [Usage] • Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning [Usage] • Determine which of the three learning styles is appropriate to a particular problem domain [Usage] • Compare and contrast each of the following techniques, providing examples of when each strategy is superior: decision trees, neural networks, and belief networks [Usage] • Evaluate the performance of a simple learning system on a real-world dataset [Usage] • Characterize the state of the art in learning theory, including its achievements and its shortcomings [Usage] • Explain the problem of overfitting, along with techniques for detecting and managing the problem [Usage]
Readings : [RN03], [KF09], [Mur12]	

Unit 8: Natural Language Processing (12)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Deterministic and stochastic grammars • Parsing algorithms – CFGs and chart parsers (e.g. CYK) – Probabilistic CFGs and weighted CYK • Representing meaning / Semantics – Logic-based knowledge representations – Semantic roles – Temporal representations – Beliefs, desires, and intentions • Corpus-based methods • N-grams and HMMs • Smoothing and backoff • Examples of use: POS tagging and morphology • Information retrieval – Vector space model ∗ TF & IDF – Precision and recall • Information extraction • Language translation • Text classification, categorization – Bag of words model	• Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each [Usage] • Simulate, apply, or implement classic and stochastic algorithms for parsing natural language [Usage] • Identify the challenges of representing meaning [Usage] • List the advantages of using standard corpora Identify examples of current corpora for a variety of NLP tasks [Usage] • Identify techniques for information retrieval, language translation, and text classification [Usage]
Readings : [Nil01], [RN03], [Pon+14]	

Unit 9: Perception and Computer Vision (12)	
Competences Expected: a,j	
Topics	**Learning Outcomes**
• Computer vision – Image acquisition, representation, processing and properties – Shape representation, object recognition and segmentation – Motion analysis • Modularity in recognition • Approaches to pattern recognition – Classification algorithms and measures of classification quality – Statistical techniques	• Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology [Usage] • List at least three image-segmentation approaches, such as thresholding, edge-based and region-based algorithms, along with their defining characteristics, strengths, and weaknesses [Usage] • Implement 2d object recognition based on contour- and/or region-based shape representations [Usage] • Provide at least two examples of a transformation of a data source from one sensory domain to another, eg, tactile data interpreted as single-band 2d images [Usage] • Implement a feature-extraction algorithm on real data, eg, an edge or corner detector for images or vectors of Fourier coefficients describing a short slice of audio signal [Usage] • Implement a classification algorithm that segments input percepts into output categories and quantitatively evaluates the resulting classification [Usage] • Evaluate the performance of the underlying feature-extraction, relative to at least one alternative possible approach (whether implemented or not) in its contribution to the classification task (8), above [Usage]
Readings : [Nil01], [RN03], [Pon+14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[De 06] L.N. De Castro. *Fundamentals of natural computing: basic concepts, algorithms, and applications.* CRC Press, 2006.

[Gol89] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning.* The MIT Press, 2009. ISBN: 0262013193.

[Mit98] M. Mitchell. *An introduction to genetic algorithms.* The MIT press, 1998.

[Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012. ISBN: 0262018020.

[Nil01] Nils Nilsson. *Inteligencia Artificial: Una nueva visión.* McGraw-Hill, 2001.

[Pon+14] Julio Ponce-Gallegos et al. *Inteligencia Artificial.* Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn), 2014.

[RN03] Stuart Russell and Peter Norvig. *Inteligencia Artifical: Un enfoque moderno.* Prentice Hall, 2003.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS291. Software Engineering I (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS113. Computer Science II. (3^{rd} Sem) • CS271. Data Management. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The aim of developing software, except for extremely simple applications, requires the execution of a well-defined development process. Professionals in this area require a high degree of knowledge of the different models and development process, so that they are able to choose the most suitable for each development project. On the other hand, the development of medium and large-scale systems requires the use of pattern and component libraries and the mastery of techniques related to component-based design

5. GOALS

- Provide the student with a theoretical and practical framework for the development of software under quality standards.
- Familiarize the student with the software modeling and construction processes through the use of CASE tools.
- Students should be able to select architectures and ad-hoc technology platforms for deployment scenarios
- Applying component-based modeling to ensure variables such as quality, cost, and time-to-market in development processes.
- Provide students with best practices for software verification and validation.

6. COMPETENCES

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Usage**)

d1) (1)

d2) (2)

d2) (2)

i1) (1)

i2) (2)

i4) (4)

i5) (**5**)

k2) (**2**)

k3) (**3**)

k4) (**4**)

k5) (**5**)

k6) (**6**)

7. TOPICS

Unit 1: Requirements Engineering (18)	
Competences Expected: i,k	
Topics	**Learning Outcomes**
• Describing functional requirements using, for example, use cases or users stories • Properties of requirements including consistency, validity, completeness, and feasibility • Software requirements elicitation • Describing system data using, for example, class diagrams or entity-relationship diagrams • Non functional requirements and their relationship to software quality • Evaluation and use of requirements specifications • Requirements analysis modeling techniques • Acceptability of certainty / uncertainty considerations regarding software / system behavior • Prototyping • Basic concepts of formal requirements specification • Requirements specification • Requirements validation • Requirements tracing	• List the key components of a use case or similar description of some behavior that is required for a system [Assessment] • Describe how the requirements engineering process supports the elicitation and validation of behavioral requirements [Assessment] • Interpret a given requirements model for a simple software system [Assessment] • Describe the fundamental challenges of and common techniques used for requirements elicitation [Assessment] • List the key components of a data model (eg, class diagrams or ER diagrams) [Assessment] • Identify both functional and non-functional requirements in a given requirements specification for a software system [Assessment] • Conduct a review of a set of software requirements to determine the quality of the requirements with respect to the characteristics of good requirements [Assessment] • Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system [Assessment] • Compare the plan-driven and agile approaches to requirements specification and validation and describe the benefits and risks associated with each [Assessment] • Use a common, non-formal method to model and specify the requirements for a medium-size software system [Assessment] • Translate into natural language a software requirements specification (eg, a software component contract) written in a formal specification language [Assessment] • Create a prototype of a software system to mitigate risk in requirements [Assessment] • Differentiate between forward and backward tracing and explain their roles in the requirements validation process [Assessment]
Readings : [ES14], [HF03]	

Unit 2: Software Design (18)	
Competences Expected: i,k	
Topics	**Learning Outcomes**
• System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion , re-use of standard structures • Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented, function oriented, service oriented • Structural and behavioral models of software designs • Design patterns • Relationships between requirements and designs: transformation of models, design of contracts, invariants • Software architecture concepts and standard architectures (e.g. client-server, n-layer, transform centered, pipes-and-filters) • The use of component desing: component selection, design, adaptation and assembly of components, component and patterns, components and objects (for example, building a GUI using a standar widget set) • Refactoring designs using design patterns • Internal design qualities, and models for them: efficiency and performance, redundacy and fault tolerance, traceability of requeriments • Measurement and analysis of design quality • Tradeoffs between different aspects of quality • Application frameworks • Middleware: the object-oriented paradigm within middleware, object request brokers and marshalling, transaction processing monitors, workflow systems • Principles of secure design and coding – Principle of least privilege – Principle of fail-safe defaults – Principle of psychological acceptability	• Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation [Familiarity] • Use a design paradigm to design a simple software system, and explain how system design principles have been applied in this design [Usage] • Construct models of the design of a simple software system that are appropriate for the paradigm used to design it [Usage] • Within the context of a single design paradigm, describe one or more design patterns that could be applicable to the design of a simple software system [Familiarity] • For a simple system suitable for a given scenario, discuss and select an appropriate design paradigm [Usage] • Create appropriate models for the structure and behavior of software products from their requirements specifications [Usage] • Explain the relationships between the requirements for a software product and its design, using appropriate models [Assessment] • For the design of a simple software system within the context of a single design paradigm, describe the software architecture of that system [Familiarity] • Given a high-level design, identify the software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, and client-server [Familiarity] • Investigate the impact of software architectures selection on the design of a simple system [Assessment] • Apply simple examples of patterns in a software design [Usage] • Describe a form of refactoring and discuss when it may be applicable [Familiarity] • Select suitable components for use in the design of a software product [Usage] • Explain how suitable components might need to be adapted for use in the design of a software product [Familiarity] • Design a contract for a typical small software component for use in a given system [Usage] • Discuss and select appropriate software architecture for a simple system suitable for a given scenario [Usage] • Apply models for internal and external qualities in designing software components to achieve an acceptable tradeoff between conflicting quality aspects [Us-

Unit 3: Software Construction (24)	
Competences Expected: i,k	
Topics	**Learning Outcomes**
• Coding practices: techniques, idioms/patterns, mechanisms for building quality programs – Defensive coding practices – Secure coding practices – Using exception handling mechanisms to make programs more robust, fault-tolerant • Coding standards • Integration strategies • Development context: “green field” vs. existing code base – Change impact analysis – Change actualization • Potential security problems in programs – Buffer and other types of overflows – Race conditions – Improper initialization, including choice of privileges – Checking input – Assuming success and correctness – Validating assumptions	• Describe techniques, coding idioms and mechanisms for implementing designs to achieve desired properties such as reliability, efficiency, and robustness [Assessment] • Build robust code using exception handling mechanisms [Assessment] • Describe secure coding and defensive coding practices [Assessment] • Select and use a defined coding standard in a small software project [Assessment] • Compare and contrast integration strategies including top-down, bottom-up, and sandwich integration [Assessment] • Describe the process of analyzing and implementing changes to code base developed for a specific project [Assessment] • Describe the process of analyzing and implementing changes to a large existing code base [Assessment] • Rewrite a simple program to remove common vulnerabilities, such as buffer overflows, integer overflows and race conditions [Assessment] • Write a software component that performs some non-trivial task and is resilient to input and run-time errors [Assessment]
Readings : [ES14], [HF03]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[ES14] Bert Bates Eric Freeman Elisabeth Robson and Kathy Sierra. *Head First Design Patterns.* 2nd. O'Reilly Media, Inc, July 2014.

[HF03] Brian Lyons Hans-Erik Eriksson Magnus Penker and Davis Fado. *UML 2 Toolkit.* 2nd. Wiley, Oct. 2003.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS2H1. User Experience (UX) (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS113. Computer Science II. (3^{rd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Language has been one of the most significant creations of humanity. From body language and gesture, through verbal and written communication, to iconic symbolic codes and others, it has made possible complex interactions Among humans and facilitated considerably the communication of information. With the invention of automatic and semi-automatic devices, including computers, The need for languages or interfaces to be able to interact with them, has gained great importance. The utility of the software, coupled with user satisfaction and increased productivity, depends on the effectiveness of the User-Computer Interface. So much so, that often the interface is the most important factor in the success and failure of any computer system. The design and implementation of appropriate Human-Computer Interfaces, which in addition to complying with the technical requirements and the transactional logic of the application, consider the subtle psychological implications, sciences and user facilities, It consumes a good part of the life cycle of a software project, and requires specialized skills, both for the construction of the same, and for the performance of usability tests.

5. GOALS

- Know and apply criteria of usability and accessibility to the design and construction of human-computer interfaces, always looking for technology to adapt to people and not people to technology.
- That the student has a vision focused on the user experience by applying appropriate conceptual and technological approaches.
- Understand how emerging technology makes possible new styles of interaction.
- Determine the basic requirements at the interface level, hardware and software for the construction of immersive environments.

6. COMPETENCES

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

d) An ability to function on multidisciplinary teams. (**Usage**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Familiarity**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Assessment**)

d) An ability to function on multidisciplinary teams. (**Usage**)

o) Improve the conditions of society by putting technology at the service of the human being. (**Familiarity**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Foundations (8)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Contexts for HCI (anything with a user interface, e.g., webpage, business applications, mobile applications, and games) • Usability heuristics and the principles of usability testing • Processes for user-centered development, e.g., early focus on users, empirical testing, iterative design • Principles of good design and good designers; engineering tradeoffs • Different measures for evaluation, e.g., utility, efficiency, learnability, user satisfaction	• Discuss why human-centered software development is important [Familiarity] • Define a user-centered design process that explicitly takes account of the fact that the user is not like the developer or their acquaintances [Familiarity] • Summarize the basic precepts of psychological and social interaction [Familiarity] • Develop and use a conceptual vocabulary for analyzing human interaction with software: affordance, conceptual model, feedback, and so forth [Familiarity]
Readings : [Dix+04], [Sto+05], [RS11]	

Unit 2: Factores Humanos (8)	
Competences Expected: b	
Topics	**Learning Outcomes**
• Cognitive models that inform interaction design, e.g., attention, perception and recognition, movement, and memory; gulfs of expectation and execution • Physical capabilities that inform interaction design, e.g., color perception, ergonomics • Accessibility, e.g., interfaces for differently-abled populations (e.g., blind, motion-impaired) • Interfaces for differently-aged population groups (e.g., children, 80+)	• Create and conduct a simple usability test for an existing software application [Familiarity]
Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Nor04]	

Unit 3: User-centered design and testing (16)	
Competences Expected: b,c	
Topics	**Learning Outcomes**
• Approaches to, and characteristics of, the design process • Functionality and usability requirements • Techniques for gathering requirements, e.g., interviews, surveys, ethnographic and contextual enquiry • Techniques and tools for the analysis and presentation of requirements, e.g., reports, personas • Task analysis, including qualitative aspects of generating task analytic models • Consideration of HCI as a design discipline – Sketching – Participatory design – Sketching – Diseño participativo • Prototyping techniques and tools, e.g., sketching, storyboards, low-fidelity prototyping, wireframes • Low-fidelity (paper) prototyping • Quantitative evaluation techniques, e.g., keystroke-level evaluation • Evaluation without users, using both qualitative and quantitative techniques, e.g., walkthroughs, GOMS, expert-based analysis, heuristics, guidelines, and standard • Evaluation with users, e.g., observation, think-aloud, interview, survey, experiment • Challenges to effective evaluation, e.g., sampling, generalization • Reporting the results of evaluations • Internationalization, designing for users from other cultures, cross-cultural	• Conduct a quantitative evaluation and discuss/report the results [Familiarity] • For an identified user group, undertake and document an analysis of their needs [Familiarity] • Discuss at least one national or international user interface design standard [Familiarity] • Explain how user-centred design complements other software process models [Familiarity] • Use lo-fi (low fidelity) prototyping techniques to gather, and report, user responses [Usage] • Choose appropriate methods to support the development of a specific UI [Assessment] • Use a variety of techniques to evaluate a given UI [Assessment] • Compare the constraints and benefits of different evaluative methods [Assessment]
Readings : [Dix+04], [Sto+05], [RS11], [Mat11], [Bux07]	

Unit 4: Designing Interaction (8)	
Competences Expected: b,c,d,o	
Topics	**Learning Outcomes**
• Principles of graphical user interfaces (GUIs) • Elements of visual design (layout, color, fonts, labeling) • Handling human/system failure • User interface standards • Presenting information: navigation, representation, manipulation • Interface animation techniques (e.g., scene graphs) • Widget classes and libraries • Internationalization, designing for users from other cultures, cross-cultural • Choosing interaction styles and interaction techniques	• Create a simple application, together with help and documentation, that supports a graphical user interface [Usage]
Readings : [Dix+04], [Sto+05], [RS11], [Joh10], [Mat11], [LS06]	

Unit 5: New Interactive Technologies (8)	
Competences Expected: o	
Topics	**Learning Outcomes**
• Choosing interaction styles and interaction techniques • Approaches to design, implementation and evaluation of non-mouse interaction – Touch and multi-touch interfaces – Shared, embodied, and large interfaces – New input modalities (such as sensor and location data) – New Windows, e.g., iPhone, Android – Speech recognition and natural language processing – Wearable and tangible interfaces – Persuasive interaction and emotion – Ubiquitous and context-aware interaction technologies (Ubicomp) – Bayesian inference (e.g. predictive text, guided pointing) – Ambient/peripheral display and interaction • Output – Sound – Stereoscopic display – Force feedback simulation, haptic devices • System architectures – Game engines – Mobile augmented reality – Flight simulators – CAVEs – Medical imaging	• Describe when non-mouse interfaces are appropriate [Familiarity] • Understand the interaction possibilities beyond mouse-and-pointer interfaces [Familiarity] • Discuss the advantages (and disadvantages) of non-mouse interfaces [Usage] • Describe the optical model realized by a computer graphics system to synthesize stereoscopic view [Familiarity] • Describe the principles of different viewer tracking technologies [Familiarity] • Determine the basic requirements on interface, hardware, and software configurations of a VR system for a specified application [Assessment]
Readings : [Dix+04], [Sto+05], [RS11], [WW11], [Mat11]	

Unit 6: Collaboration and communication (8)	
Competences Expected: d,o	
Topics	**Learning Outcomes**
• Asynchronous group communication, e.g., e-mail, forums, social networks • Social media, social computing, and social network analysis • Online collaboration, 'smart' spaces, and social coordination aspects of workflow technologies • Online communities • Software characters and intelligent agents, virtual worlds and avatars • Social psychology	• Describe the difference between synchronous and asynchronous communication [Familiarity] • Compare the HCI issues in individual interaction with group interaction [Familiarity] • Discuss several issues of social concern raised by collaborative software [Usage] • Discuss the HCI issues in software that embodies human intention [Assessment]
Readings : [Dix+04], [Sto+05], [RS11]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Bux07] Bill Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., 2007.

[Dix+04] Alan Dix et al. *Human-computer Interaction*. 3 ed. Prentice-Hall, Inc, 2004.

[Joh10] Jeff Johnson. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. 3 ed. Morgan Kaufmann Publishers Inc., 2010.

[LS06] M. Leavitt and B. Shneiderman. *Research-Based Web Design & Usability Guidelines*. Health and Human Services Dept, 2006.

[Mat11] Lukas Mathis. *Designed for Use: Create Usable Interfaces for Applications and the Web*. Pragmatic Bookshelf, 2011.

[Nor04] Donald A. Norman. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Book, 2004.

[RS11] Y. Rogers and J Sharp H. & Preece. *Interaction Design: Beyond Human-Computer Interaction*. 3 ed. John Wiley and Sons Ltd, 2011.

[Sto+05] D. Stone et al. *User Interface Design and Evaluation*. Morgan Kaufmann Series in Interactive Technologies, 2005.

[WW11] D. Wigdor and D. Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann Publishers Inc, 2011.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CB111. Computational Physics (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	MA100. Mathematics I. (1st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Física I es un curso que le permitirá al estudiante entender las leyes de física de macropartículas y micropartículas considerado desde un punto material hasta un sistemas de partículas; debiéndose tener en cuenta que los fenómenos aquí estudiados se relacionan a la física clásica: Cinemática, Dinámica, Trabajo y Energía; además se debe asociar que éstos problemas deben ser resueltos con algoritmos computacionales.

Poseer capacidad y habilidad en la interpretación de problemas clásicos con condiciones de frontera reales que contribuyen en la elaboración de soluciones eficientes y factibles en diferentes áreas de la Ciencia de la Computación.

5. GOALS

- Conocer los principios básicos de los fenómenos que gobiernan la física clásica.
- Aplicar los principios básicos a situaciones específicas y poder asociarlos con situaciones reales.
- Analizar algunos de los fenómenos físicos así como su aplicación a situaciones reales.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Familiarity**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: (6)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Análisis dimensional. • Vectores. Propiedades. Operaciones. • Caso práctico: Estimación de fuerzas.	• Entender y trabajar con las magnitudes físicas del SI.[Usage] • Abstraer de la naturaleza los conceptos físicos rigurosos y representarlos en modelos vectoriales.[Usage] • Entender y aplicar los conceptos vectoriales a problemas físicos reales.[Usage]
Readings : [Bur06], [Res07], [Ser09], [Tip09]	

Unit 2: (6)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Primera y tercera Ley de Newton. • Diagrama de cuerpo libre. • Primera condición de equilibrio. • Caso práctico: Estimación de la fuerza humana. • Segunda condición de equilibrio. • Torque. • Casos prácticos: Aplicaciones en dispositivos mecánicos. • Fricción.	• Conocer los conceptos que rigen la primera Ley y tercera Ley de Newton. • Conocer y aplicar los conceptos de la primera y segunda condición de equilibro. • Capacidad para resolver problemas de casos prácticos. • Entender el concepto de fricción y resolver problemas.
Readings : [Bur06], [Res07], [Ser09], [Tip09]	

Unit 3: (6)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• Posición, Velocidad, Aceleración. • Gráficas de movimiento. • Casos prácticos: Representación gráfica de movimiento utilizando Excel. • Movimiento circular. • Velocidad angular y velocidad tangencial. • Mecanismos rotativos. • Caso práctico: Operación de la caja de cambios de un automóvil.	• Poder determinar la posición, velocidad y aceleración de un cuerpo. • Conocer el concepto de composición de movimientos y saberlo aplicar, en la descripción de un movimiento circular. • Conocer el significado de las componentes tangencial y normal de la aceleración y saberlas calcular en un instante determinado. • Utilizar excel para el procesamiento de datos experimentales.
Readings : [Bur06], [Res07], [Ser09], [Tip09]	

Unit 4: (6)	
Competences Expected: C1	
Topics	**Learning Outcomes**
• Segunda Ley de Newton. • Fuerza y movimiento. • Momento de inercia.	• Aplicar las leyes de Newton en la solución de problemas. • Describir las diversas interacciones por sus correspondientes fuerzas. • Determinar el momento de inercia de un cuerpo usando un método dinámico
Readings : [Bur06], [Res07], [Ser09], [Tip09]	

Unit 5: (6)	
Competences Expected: C20	
Topics	**Learning Outcomes**
• Trabajo. • Fuerzas constantes. • Fuerzas variables. • Potencia. • Caso práctico: Estimación de la potencia de una planta hidroeléctrica.	• Comprender el concepto de Trabajo. • Comprender y aplicar el concepto de Potencia a la resolución de problemas. • Resolver problemas.
Readings : [Bur06], [Res07], [Ser09], [Tip09]	

Unit 6: (6)	
Competences Expected: C24	
Topics	**Learning Outcomes**
• Tipos de energía. • Conservación de la energía. • Dinámica de un sistema de partículas. • Colisiones.	• Conocer los tipos de energía que existen. • Aplicar el principio de conservación de la energía mecánica a distintas situaciones, diferenciando aquellas en las que la energía total no se mantiene constante. • Aplicar los principios de conservación del momento lineal y de la energía a un sistema aislado de dos o más partículas interactuantes.
Readings : [Bur06], [Res07], [Ser09], [Tip09]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Bur06] S. Burbano. *Física General*. Alfaomega, 2006.

[Res07] D. Resnik R. y Halliday. *Física*. 5th. Vol. 1. Patria, 2007.

[Ser09] J.W. Serway R. A. y Jewett. *Física para Ciencias e Ingeniería con Física Moderna*. 7th. Vol. 1. Cengage Learning, 2009.

[Tip09] G. Tipler P. y Mosca. *Física para la ciencia y la tecnología*. 7th. Vol. 1. Reverte, 2009.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

ID203. Technical and professional English V (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	-
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	ID202. English IV. (4^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

A fundamental part of the integral formation of a professional is the ability to communicate in a foreign language in addition to the native language itself. It not only broadens its cultural horizon but also allows a more humane and comprehensive view of life. In the case of foreign languages, undoubtedly English is the most practical because it is spoken around the world. There is no country where it is not spoken. In careers related to tourist services, English is perhaps the most important practical tool that the student must master from the outset as part of his / her integral education

5. GOALS

- Increase the ability and fluency of speaking and understanding the English language.
- That the students interact with greater emphasis in the creation of dialogues.

6. COMPETENCES

f) An ability to communicate effectively. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: It's a wonderful world (0)	
Competences Expected: 5	
Topics	**Learning Outcomes**
• Auxiliary verbs • Verb tenses • Negative Questions and Prayers • Short answers • Word formation • Colloquial expressions • Error correction	• At the end of the first unit, each student, understanding the grammar of auxiliaries and different types of sentences, is able to express a greater number of expressions of time and also use prepositions to describe varied places and times. He is also able to analyze and express ideas about word formation.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 2: Happiness! (0)	
Competences Expected: 5	
Topics	**Learning Outcomes**
• Simple present • Present continuous • Passive Voice in Present • Verbs for sports and free time • Types of numbers and • Inventions / Modern World • Corrección de errores	• At the end of the second unit, students have identified how to express sports and leisure activities. It uses all kinds of numerical expressions. Express situations and states related to present forms. Explain and apply vocabulary of outdoor activities.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 3: Telling tales! (0)	
Competences Expected: 5	
Topics	**Learning Outcomes**
• Simple Past Time • Past Continuous • Passive Voice in Past • Vocabulary of Art and Literature • Expressions to give and ask opinions • Stories and stories	• At the end of the third unit, students having recognized the characteristics of the past passive forms, they use these make descriptions of various types. Describe art and literature and give indications of opinion. They will use conjunctions to unite type ideas.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 4: Doing the right thing! (0)	
Competences Expected: 5	
Topics	**Learning Outcomes**
• Mode Auxiliary Verbs I. • Affirmative, Negative and Modals Questions • Use of nationalities and other adjectives • Expressions of orders and offers • Guide to Good Manners • Form Fill • Phonetic symbols	• At the conclusion of the fourth unit, the students having identified the idea of expressing ideas of modes of actions that happen at the moment or that are related at any time, structure sentences in the Present. They express ideas of nationalities and make requests and offers varied.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 5: On the move! (0)	
Competences Expected: 5	
Topics	**Learning Outcomes**
• Future with Will • Future Time Prayers with going to • Use of might for future • Climate Expressions • Vocabulary of the climate • Expressions for hotels and transportation • E-mails	• At the end of the fifth unit, students, from the understanding of future time, will elaborate sentences using the necessary elements. They will also assimilate the need to express ideas of the climate. They will acquire vocabulary to describe use of public transportation. Expressions will be presented to order at hotels.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

Unit 6: I just love it! (0)	
Competences Expected: 5	
Topics	**Learning Outcomes**
• Questions with Shapes Like • Patrones Verbales II • Vocabulario de Comida, Lugares y ocupaciones • Palabras que van unidas en contexto • Expresiones para vistas y sonidos • Composición de Impresiones personales	• At the end of the sixth unit, students having learned the basics of structuring questions with like and with verbal patterns work applied to appropriate contexts. They emphasize the difference between meals, places and people. Describe sights and sounds. They use expressions to compare daily life in different places. They assume the idea of different lifestyles.
Readings : [SJ02a], [SJ02c], [SJ02b], [Cam06], [Mac99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Cam06] Cambridge. *Diccionario Inglés-Espanol Cambridge*. Editorial Oxford, 2006.

[Mac99] James MacGrew. *Focus on Grammar Basic*. Editorial Oxford, 1999.

[SJ02a] Liz Soars and John. *American Headway N 3 Student Book*. Editorial Oxford, 2002.

[SJ02b] Liz Soars and John. *American Headway N 3 Teachers Book*. Editorial Oxford, 2002.

[SJ02c] Liz Soars and John. *American Headway N 3 Work Book*. Editorial Oxford, 2002.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS292. Software Engineering II (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS291. Software Engineering I. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The topics of this course extend the ideas of software design and development from the introduction sequence to programming to encompass the problems encountered in large-scale projects. It is a broader and more complete view of Software Engineering appreciated from a Project point of view.

5. GOALS

- Enable students to be part of and define software development teams facing real-world problems.
- familiarize the students with the process of administering a software project in such a way as to be able to create, improve and use tools and metrics that allow them to carry out the estimation and monitoring of a software project
- Create, evaluate and execute a test plan for medium-sized code segments, Distinguish between different types of tests, lay the foundation for creating, improve test procedures and tools for these purposes
- Select with justification an appropriate set of tools to support the development of a range of software products.
- Create, improve and use existing patterns for software maintenance. Disclose features and design patterns for software reuse.
- Identify and discuss different specialized systems, create, improve and use specialized standards for the design, implementation, maintenance and testing of specialized systems.

6. COMPETENCES

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

d) An ability to function on multidisciplinary teams. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

f) An ability to communicate effectively. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Assessment**)

c1) (**1**)

c3) (**3**)

c4) (4)

d1) (1)

d2) (2)

d2) (2)

i1) (1)

i2) (2)

i4) (4)

i5) (5)

k2) (2)

k3) (3)

k4) (4)

k5) (5)

k6) (6)

7. TOPICS

Unit 1: Tools and Environments (12)	
Competences Expected: c,f,i	
Topics	**Learning Outcomes**
• Software configuration management and version control • Release management • Requierements analysis and desing modeling tools • Testing tools including static and dynamic analysis tools • Programming enviroments that automate parts of program construction pocesses (e.g., automated builds) – Continuous integration • Tool integration concepts and mechanisms	• Software configuration management and version control [Usage] • Release management [Usage] • Requierements analysis and desing modeling tools [Usage] • Testing tools including static and dynamic analysis tools [Usage] • Programming enviroments that automate parts of program construction pocesses (e.g., automated builds) – Continuous integration [Usage] • Tool integration concepts and mechanisms [Usage]
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 2: Software Verification and Validation (12)	
Competences Expected: c,f,i	
Topics	**Learning Outcomes**
• Verification and validation concepts • Inspections, reviews, audits • Testing types, including human computer interface, usability, reliability, security, conformance to specification • Testing fundamentals – Unit, integration, validation, and system testing – Test plan creation and test case generation – Black-box and white-box testing techniques – Regression testing and test automation • Defect tracking • Limitations of testing in particular domains, such as parallel or safety-critical systems • Static approaches and dynamic approaches to verification • Test-driven development • Validation planning; documentation for validation • Object-oriented testing; systems testing • Verification and validation of non-code artifacts (documentation, help files, training materials) • Fault logging, fault tracking and technical support for such activities • Fault estimation and testing termination including defect seeding	• Distinguish between program validation and verification [Usage] • Describe the role that tools can play in the validation of software [Usage] • Undertake, as part of a team activity, an inspection of a medium-size code segment [Usage] • Describe and distinguish among the different types and levels of testing (unit, integration, systems, and acceptance) [Usage] • Describe techniques for identifying significant test cases for integration, regression and system testing [Usage] • Create and document a set of tests for a medium-size code segment [Usage] • Describe how to select good regression tests and automate them [Usage] • Use a defect tracking tool to manage software defects in a small software project [Usage] • Discuss the limitations of testing in a particular domain [Usage] • Evaluate a test suite for a medium-size code segment [Usage] • Compare static and dynamic approaches to verification [Usage] • Identify the fundamental principles of test-driven development methods and explain the role of automated testing in these methods [Usage] • Discuss the issues involving the testing of object-oriented software [Usage] • Describe techniques for the verification and validation of non-code artifacts [Usage] • Describe approaches for fault estimation [Usage] • Estimate the number of faults in a small software application based on fault density and fault seeding [Usage] • Conduct an inspection or review of software source code for a small or medium sized software project [Usage]
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 3: Software Evolution (12)	
Competences Expected: c,f,i	
Topics	**Learning Outcomes**
• Software development in the context of large, preexisting code bases – Software change – Concerns and concernlocation – Refactoring • Software evolution • Characteristics of maintainable software • Reengineering systems • Software reuse – Code segments – Libraries and frameworks – Components – Product lines	• Identify the principal issues associated with software evolution and explain their impact on the software lifecycle [Usage] • Estimate the impact of a change request to an existing product of medium size [Usage] • Use refactoring in the process of modifying a software component [Usage] • Discuss the challenges of evolving systems in a changing environment [Usage] • Outline the process of regression testing and its role in release management [Usage] • Discuss the advantages and disadvantages of different types of software reuse [Usage]
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

Unit 4: Software Project Management (12)	
Competences Expected: c,f,i	
Topics	**Learning Outcomes**
• Team participation – Team processes including responsabilities for task, meeting structure, and work schedule – Roles and responsabilities in a software team – Team conflict resolution – Risks associated with virtual teams (communication, perception, structure) • Effort estimation (at the personal level) • Risk – The role of risk in the lifecycle – Risk categories including security, safety, market, financial, technology, people, quality, structure and process • Team management – Team organization and decision-making – Role identification and assigment – Individual and team performance assessment • Project management – Scheduling and tracking – Project management tools – Cost/benefit analysis • Software measurement and estimation techniques • Software quality assurance and the role of measurements • Risk – Risk identification and management – Risk analysis and evaluation – Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking) – Risk planning • System-wide approach to risk including hazards associated with tools	• Discuss common behaviors that contribute to the effective functioning of a team [Usage] • Create and follow an agenda for a team meeting [Usage] • Identify and justify necessary roles in a software development team [Usage] • Understand the sources, hazards, and potential benefits of team conflict [Usage] • Apply a conflict resolution strategy in a team setting [Usage] • Use an ad hoc method to estimate software development effort (eg, time) and compare to actual effort required [Usage] • List several examples of software risks [Usage] • Describe the impact of risk in a software development lifecycle [Usage] • Describe different categories of risk in software systems [Usage] • Demonstrate through involvement in a team project the central elements of team building and team management [Usage]
Readings : [Pre04], [Blu92], [Sch04], [WK00], [Key04], [WA02], [PS01], [Sch04], [Mon96], [Amb01], [Con00], [Oqu03]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Amb01] Vincenzo Ambriola. *Software Process Technology*. Springer, July 2001.

[Blu92] Bruce I. Blum. *Software Engineering: A Holistic View*. 7th. Oxford University Press US, May 1992.

[Con00] R Conradi. *Software Process Technology*. Springer, Mar. 2000.

[Key04] Jessica Keyes. *Software Configuration Management*. CRC Press, Feb. 2004.

[Mon96] Carlo Montangero. *Software Process Technology*. Springer, Sept. 1996.

[Oqu03] Flavio Oquendo. *Software Process Technology*. Springer, Sept. 2003.

[Pre04] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. 6th. McGraw-Hill, Mar. 2004.

[PS01] John W. Priest and Jose M. Sanchez. *Product Development and Design for Manufacturing*. Marcel Dekker, Jan. 2001.

[Sch04] Stephen R Schach. *Object-Oriented and Classical Software Engineering*. McGraw-Hill, Jan. 2004.

[WA02] Daniel R. Windle and L. Rene Abreo. *Software Requirements Using the Unified Process*. Prentice Hall, Aug. 2002.

[WK00] Yingxu Wang and Graham King. *Software Engineering Processes: Principles and Applications*. CRC Press, Apr. 2000.

Ministry of Education (MINEDU)

Sillabus 2021-I

1. COURSE

CS311. Competitive Programming (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS212. Analysis and Design of Algorithms. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Competitive Programming combines problem-solving challenges with the fun of competing with others. It teaches participants to think faster and develop problem-solving skills that are in high demand in the industry. This course will teach you to solve algorithmic problems quickly by combining theory of algorithms and data structures with practice solving problems.

5. GOALS

- That the student uses techniques of data structures and complex algorithms..
- That the student apply the concepts learned for the application on a real problem.
- That the student investigate the possibility of creating a new algorithm and / or new technique to solve a real problem.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

a4) (4)

b1) (1)

7. TOPICS

Unit 1: Introduction (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Introduction to Competetive Programming • Computacional model • Runtime and space complexity • Recurrence and recursion • Divide and conquer	• Identify and learn how to use the resources in the Random Access Machine (RAM) computacional model. [Usage] • Compute the runtime and space complexity for written algorithms. [Usage] • Compute the recurrence relations for recursive algorithms. [Usage] • Solve problems related to searching and sorting. [Usage] • Learning to select the right algorithms for divide-and-conquer problems. [Usage] • Design new algorithms for real-world problem solving.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 2: Data structure (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Arrays and strings problems • Linked lists problems • Stacks and queues problems • Trees problems • Hash tables problems • Heaps problems	• Recognize different data structures, their complexities, uses and restrictions.[Usage] • Identify the type of data structure appropriate to the resolution of the problem. [Usage] • Recognize types of problems associated with operations on data structures such as searching, inserting, deleting and updating.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 3: Algorithmic Design Paradigms (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Brute force • Divide and conquer • Backtracking • Greedy • Dynamic Programming	• Learning the different algorithhmic design paradigms.[Usage] • Learning to select the right algorithms for different problems applying different algorithhmic design paradigms.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 4: Graphs (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Graphs transversal • Graphs aplications • Shortest path • Networks and flows	• Identify problems classified as graph problems. [Usage] • Learn how to select the right algorithms for network problems (transversal, MST, shortest-path, network and flows). [Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 5: Advanced topics (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Number theory • Probabilities and combinations • String algorithms (tries, string hashing, z-algorithm) • Geometric algorithms	• Learning to select the right algorithms for problems in number theory and mathematics as they are important in competitive programming. [Usage] • Learning to select the right algorithms for problems about probabilities and combinations, strings and computational geometry. [Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

Unit 6: Domain specific problems (20)	
Competences Expected: a,b,h	
Topics	**Learning Outcomes**
• Latency and throughput • Parallelism • Networks • Storage • High availability • Caching • Proxies • Load balancers • Key-value stores • Replicating and sharing • Leader election • Rate limiting • Logging and monitoring	• Learning to design systems for different domain-specific problems by applying knowledge about networks, distributed computing, high availability, storage and system architecture.[Usage]
Readings : [Cor+09], [Hal13], [Kul19], [Mig03], [Laa17], [ALP12]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[ALP12] A. Aziz, T.H. Lee, and A. Prakash. *Elements of Programming Interviews: The Insiders' Guide*. ElementsOfProgrammingInterviews.com, 2012. ISBN: 9781479274833. URL: `https://books.google.com.pe/books?id=y6FLBQAAQBAJ`.

[Cor+09] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, 2009.

[Hal13] Steven Halim. *Competitive Programming*. 3 rd. Lulu, 2013.

[Kul19] Alexander S. Kulikov. *Learning Algorithms Through Programming and Puzzle Solving*. Active Learning Technologies, 2019.

[Laa17] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Stringer, 2017.

[Mig03] Steve Skiena Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer, May 2003. ISBN: 978-0387001630.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS312. Advanced Data Structures (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS212. Analysis and Design of Algorithms. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Algorithms and data structures are a fundamental part of computer science that allow us to organize information more efficiently, so it is important for every professional in the area to have a solid background in this regard.

In the course of advanced data structures our goal is for the student to know and analyze complex structures, such as Multidimensional Access Methods, Spatio-Temporal Access Methods and Metric Access Methods, Compact Data Structures, etc.

5. GOALS

- That the student understands, designs, implements, applies and Propose innovative data structures to solve problems related to the handling of multidimensional data, retrieval of information by similarity, search engines and other computational problems.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Familiarity**)

a) An ability to apply knowledge of mathematics, science. (**Familiarity**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Familiarity**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Técnicas Básicas de Implementación de Estructuras de Datos (16)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Structured Programming • Object-oriented programming • Abstract Data Types • Independence of the user programming language of the structure • Platform Independence • Concurrency control • Data Protection • Encapsulation levels (struct, class, namespace, etc)	• That the student understands the basic differences that involve the different techniques of implementation of data structures[Usage] • That the student analyze the advantages and disadvantages of each of the existing techniques[Usage]
Readings : [Cua+04], [Knu07a], [Knu07b], [Gam+94], [Bjö18], [Dav18]	

Unit 2: Métodos de Acceso Multidimensionales (16)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Access Methods for Point Data • Access Methods for non-point data • Problems with dimension enhancement	• That the student understands to know and implement some Access Methods for multidimensional data and temporal space[Usage] • That the student understands the potential of these Access Methods in the future of commercial databases[Usage]
Readings : [Sam06], [Gü98]	

Unit 3: Métodos de Acceso Métrico (20)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Metric Access Methods for discrete distances • Metric Access Methods for Continuous Distances	• That the student understands to know and implement some methods of metric access[Usage] • That the student understands the importance of these Access Methods for Information Retrieval by similarity[Usage]
Readings : [Sam06], [Chá+01], [Tra+00], [Zez+07]	

Unit 4: Métodos de Acceso Aproximados (20)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Space Filling Curves • Locality Sensitive Hashing	• That the student understands to know and implement some approximate access methods[Usage] • That the student understands the importance of these Access Methods for Information Retrieval by Similarity in environments where Scalability is a very important factor [Usage]
Readings : [Sam06], [PI06], [Zez+07]	

Unit 5: Seminarios (8)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Access Methods Temporary Space • Generic Data Structures	• That the student can discuss the latest advances in access methods for different domains of knowledge [Usage]
Readings : [Sam06], [Nav16], [Chá+01]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Bjö18] Stefan Björnander. *C++17 By Example: Practical projects to get you up and running with C++17*. Packt Publishing, Feb. 2018.

[Chá+01] E. Chávez et al. “Proximity Searching in Metric Spaces”. In: *ACM Computing Surveys* 33.3 (Sept. 2001), pp. 273–321.

[Cua+04] Ernesto Cuadros-Vargas et al. “Implementing data structures: An incremental approach”. http://socios.spc.org.pe/ecuadros/cursos/pdfs/. 2004.

[Dav18] Doug Gregor David Vandevoorde Nicolai M. Josuttis. *C++ Templates: The Complete Guide*. Addison-Wesley Professional, Sept. 2018. URL: `http://informit.com/aw`.

[Gam+94] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Computing Series. ISBN-10: 0201633612. Addison-Wesley Professional, Nov. 1994.

[Gü98] Volker Gaede and Oliver ünther. “Multidimensional Access Methods”. In: *ACM Computing Surveys* 30.2 (1998), pp. 170–231.

[Knu07a] Donald Ervin Knuth. *The Art of Computer Programming, Fundamental Algorithms.* 3rd. Vol. I. 0-201-89683-4. Addison-Wesley, Feb. 2007.

[Knu07b] Donald Ervin Knuth. *The Art of Computer Programming, Sorting and Searching.* 2nd. Vol. II. 0-201-89685-0. Addison-Wesley, Feb. 2007.

[Nav16] Gonzalo Navarro. *Compact Data Structures.* Cambridge University Press, 2016. ISBN: 978-1107152380.

[PI06] Trevor Darrell PGregory Shakhnarovich and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice.* 1st. ISBN 0-262-19547-X. MIT Press, Mar. 2006.

[Sam06] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures.* Illustrated. Elsevier/Morgan Kaufmann, Aug. 2006. ISBN: 9780123694461. URL: `http://books.google.com.pe/books?id=vO-NRRKHG84C`.

[Tra+00] C. Traina Jr et al. “Slim-Trees: High Performance Metric Trees Minimizing Overlap between Nodes”. In: *Advances in Database Technology - EDBT 2000, 6th International Conference on Extending Database Technology.* Vol. 1777. Lecture Notes in Computer Science. Konstanz, Germany: Springer, Mar. 2000, pp. 51–65.

[Zez+07] Pavel Zezula et al. *Similarity Search: The Metric Space Approach.* 1st. ISBN-10: 0387291466. Springer, Nov. 2007.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS393. Information systems (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS291. Software Engineering I. (5^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Analyze techniques for the correct implementation of scalable, robust, reliable and efficient information systems in organizations.

5. GOALS

- Implement correctly (scalable, robust, reliable and efficient) Information Systems in organizations.

6. COMPETENCES

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

k) Apply the principles of development and design in the construction of software systems of variable complexity. (**Assessment**)

c2) (**2**)

i1) (**1**)

k1) (**1**)

7. TOPICS

Unit 1: Introduction (15)	
Competences Expected: c,i	
Topics	**Learning Outcomes**
• Introduction to information management. • Software for information management. • Technology for information management.	• Correctly apply technology for information management [Assessment]
Readings : [Som17], [PM15], [LL17]	

Unit 2: Strategy (15)	
Competences Expected: i,k	
Topics	**Learning Outcomes**
• Strategy for information management. • Strategy for knowledge management • Strategy for information system.	• Apply and evaluate correctly management strategies [Assessment]
Readings : [Som17], [PM15]	

Unit 3: Implementation (15)	
Competences Expected: c,i,k	
Topics	**Learning Outcomes**
• Management Information Systems Development. • Change management • Information Architecture	• Implement and correctly evaluate implementation strategies [Assessment]
Readings : [Som17], [PM15]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[LL17] Kenneth C. Laudon and Jane P. Laudon. *Management Information Systems: Managing the Digital Firm*. 15th. Pearson, Mar. 2017.

[PM15] Roger S. Pressman and Bruce Maxim. *Software Engineering: A Practitioner's Approach*. 8th. McGraw-Hill, Jan. 2015.

[Som17] Ian Sommerville. *Software Engineering*. 10th. Pearson, Mar. 2017.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS3I1. Computer Security (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	3
2.2 Theory Hours	:	1 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS231. Networking and Communication. (5th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

Nowadays, information is one of the most valuable assets in any organization. This course is oriented to be able to provide the student with the security elements oriented to protect the Information of the organization and mainly to be able to foresee the possible problems related to this heading. This subject involves the development of a preventive attitude on the part of the student in all areas related to software development.

5. GOALS

- Discuss at an intermediate intermediate level the fundamentals of Computer Security.
- Provide different aspects of the malicious code.
- That the student knows the concepts of cryptography and security in computer networks.
- Discuss and analyze together with the student the aspects of Internet Security.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

e) Understand correctly the professional, ethical, legal, security and social implications of the profession. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Assessment**)

h) A recognition of the need for, and an ability to engage in life-long learning. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

■NoSpecificOutcomes■

7. TOPICS

Unit 1: Foundational Concepts in Security (25)	
Competences Expected: a,g	
Topics	**Learning Outcomes**
• CIA (Confidentiality, Integrity, Availability) • Concepts of risk, threats, vulnerabilities, and attack vectors • Authentication and authorization, access control (mandatory vs. discretionary) • Concept of trust and trustworthiness • Ethics (responsible disclosure)	• Analyze the tradeoffs of balancing key security properties (Confidentiality, Integrity, Availability) [Familiarity] • Describe the concepts of risk, threats, vulnerabilities and attack vectors (including the fact that there is no such thing as perfect security) [Familiarity] • Explain the concepts of authentication, authorization, access control [Familiarity] • Explain the concept of trust and trustworthiness [Familiarity] • Recognize that there are important ethical issues to consider in computer security, including ethical issues associated with fixing or not fixing vulnerabilities and disclosing or not disclosing vulnerabilities [Familiarity]
Readings : [WL14]	

<table>
<tr><th colspan="2">Unit 2: Principles of Secure Design (25)</th></tr>
<tr><th colspan="2">Competences Expected: g,a,e,h</th></tr>
<tr><th>Topics</th><th>Learning Outcomes</th></tr>
<tr><td>• Least privilege and isolation
• Fail-safe defaults
• Open design
• End-to-end security
• Defense in depth (e.g., defensive programming, layered defense)
• Security by design
• Tensions between security and other design goals
• Complete mediation
• Use of vetted security components
• Economy of mechanism (reducing trusted computing base, minimize attack surface)
• Usable security
• Security composability
• Prevention, detection, and deterrence</td><td>• Describe the principle of least privilege and isolation as applied to system design [Familiarity]
• Summarize the principle of fail-safe and deny-by-default [Familiarity]
• Discuss the implications of relying on open design or the secrecy of design for security. [Familiarity]
• Explain the goals of end-to-end data security [Familiarity]
• Discuss the benefits of having multiple layers of defenses [Familiarity]
• For each stage in the lifecycle of a product, describe what security considerations should be evaluated. [Familiarity]
• Describe the cost and tradeoffs associated with designing security into a product [Familiarity]
• Describe the concept of mediation and the principle of complete mediation [Familiarity]
• Be aware of standard components for security operations, instead of re-inventing fundamentals operations [Familiarity]
• Explain the concept of trusted computing including trusted computing base and attack surface and the principle of minimizing trusted computing base [Familiarity]
• Discuss the importance of usability in security mechanism design [Familiarity]
• Describe security issues that arise at boundaries between multiple components. [Familiarity]
• Identify the different roles of prevention mechanisms and detection/deterrence mechanisms [Familiarity]</td></tr>
<tr><td colspan="2"><b>Readings :</b> [WL14]</td></tr>
</table>

Unit 3: Defensive Programming (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**
• Input validation and data sanitization • Choice of programming language and type-safe languages • Examples of input validation and data sanitization errors – Buffer overflows – Integer errors – SQL injection – XSS vulnerability • Race conditions • Correct handling of exceptions and unexpected behaviors • Correct usage of third-party components • Effectively deploying security updates • Information flow control • Correctly generating randomness for security purposes • Mechanisms for detecting and mitigating input and data sanitization errors • Fuzzing • Static analysis and dynamic analysis • Program verification • Operating system support (e.g., address space randomization, canaries) • Hardware support (e.g, DEP, TPM)	• Explain why input validation and data sanitization is necessary in the face of adversarial control of the input channel. [Usage] • Explain why you might choose to develop a program in a type-safe language like Java, in contrast to an unsafe programming language like C/C++ [Usage] • Classify common input validation errors, and write correct input validation code [Usage] • Demonstrate using a high-level programming language how to prevent a race condition from occurring and how to handle an exception [Usage] • Demonstrate the identification and graceful handling of error conditions [Familiarity] • Explain the risks with misusing interfaces with third-party code and how to correctly use third-party code [Familiarity] • Discuss the need to update software to fix security vulnerabilities and the lifecycle management of the fix [Familiarity]
Readings : [WL14]	

Unit 4: Threats and Attacks (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**
• Attacker goals, capabilities, and motivations (such as underground economy, digital espionage, cyberwarfare, insider threats, hacktivism, advanced persistent threats) • Examples of malware (e.g., viruses, worms, spyware, botnets, Trojan horses or rootkits) • Denial of Service (DoS) and Distributed Denial of Service (DDoS) • Social engineering (e.g., phishing) • Attacks on privacy and anonymity • Malware/unwanted communication such as covert channels and steganography	• Describe likely attacker types against a particular system [Familiarity] • Discuss the limitations of malware countermeasures (eg, signature-based detection, behavioral detection) [Familiarity] • Identify instances of social engineering attacks and Denial of Service attacks [Familiarity] • Discuss how Denial of Service attacks can be identified and mitigated [Familiarity] • Describe risks to privacy and anonymity in commonly used applications [Familiarity] • Discuss the concepts of covert channels and other data leakage procedures [Familiarity]
Readings : [WL14]	

Unit 5: Network Security (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**
• Network specific threats and attack types (e.g., denial of service, spoofing, sniffing and traffic redirection, man-in-the-middle, message integrity attacks, routing attacks, and traffic analysis) • Use of cryptography for data and network security • Architectures for secure networks (e.g., secure channels, secure routing protocols, secure DNS, VPNs, anonymous communication protocols, isolation) • Defense mechanisms and countermeasures (e.g., network monitoring, intrusion detection, firewalls, spoofing and DoS protection, honeypots, tracebacks) • Security for wireless, cellular networks • Other non-wired networks (e.g., ad hoc, sensor, and vehicular networks) • Censorship resistance • Operational network security management (e.g., configure network access control)	• Describe the different categories of network threats and attacks [Familiarity] • Describe the architecture for public and private key cryptography and how PKI supports network security [Familiarity] • Describe virtues and limitations of security technologies at each layer of the network stack [Familiarity] • Identify the appropriate defense mechanism(s) and its limitations given a network threat [Usage]
Readings : [WL14]	

Unit 6: Cryptography (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**
• Basic Cryptography Terminology covering notions pertaining to the different (communication) partners, secure/unsecure channel, attackers and their capabilities, encryption, decryption, keys and their characteristics, signatures • Cipher types (e.g., Caesar cipher, affine cipher) together with typical attack methods such as frequency analysis • Public Key Infrastructure support for digital signature and encryption and its challenges • Symmetric key cryptography – Perfect secrecy and the one time pad – Modes of operation for semantic security and authenticated encryption (e.g., encrypt-then-MAC, OCB, GCM) – Message integrity (e.g., CMAC, HMAC) • Public key cryptography: – Trapdoor permutation, e.g., RSA – Public key encryption, e.g., RSA encryption, EI Gamal encryption – Digital signatures – Public-key infrastructure (PKI) and certificates – Hardness assumptions, e.g., Diffie-Hellman, integer factoring • Authenticated key exchange protocols, e.g., TLS • Cryptographic primitives: – pseudo-random generators and stream ciphers – block ciphers (pseudo-random permutations), e.g., AES – pseudo-random functions – hash functions, e.g., SHA2, collision resistance – message authentication codes – key derivations functions	• Describe the purpose of Cryptography and list ways it is used in data communications [Familiarity] • Define the following terms: Cipher, Cryptanalysis, Cryptographic Algorithm, and Cryptology and describe the two basic methods (ciphers) for transforming plain text in cipher text [Familiarity] • Discuss the importance of prime numbers in cryptography and explain their use in cryptographic algorithms [Familiarity] • Illustrate how to measure entropy and how to generate cryptographic randomness [Usage] • Use public-key primitives and their applications [Usage] • Explain how key exchange protocols work and how they fail [Familiarity] • Discuss cryptographic protocols and their properties [Familiarity]
Readings : [WL14]	

Unit 7: Web Security (25)	
Competences Expected: a,g	
Topics	**Learning Outcomes**
• Web security model – Browser security model including same-origin policy – Client-server trust boundaries, e.g., cannot rely on secure execution in the client • Session management, authentication – Single sign-on – HTTPS and certificates • Application vulnerabilities and defenses – SQL injection – XSS – CSRF • Client-side security – Cookies security policy – HTTP security extensions, e.g. HSTS – Plugins, extensions, and web apps – Web user tracking • Server-side security tools, e.g. Web Application Firewalls (WAFs) and fuzzers	• Describe the browser security model including same-origin policy and threat models in web security [Familiarity] • Discuss the concept of web sessions, secure communication channels such as TLS and importance of secure certificates, authentication including single sign-on such as OAuth and SAML [Familiarity] • Investigate common types of vulnerabilities and attacks in web applications, and defenses against them [Familiarity] • Use client-side security capabilities [Usage]
Readings : [WL14]	

Unit 8: Platform Security (25)	
Competences Expected: b,e,i	
Topics	**Learning Outcomes**
• Code integrity and code signing • Secure boot, measured boot, and root of trust • Attestation • TPM and secure co-processors • Security threats from peripherals, e.g., DMA, IOMMU • Physical attacks: hardware Trojans, memory probes, cold boot attacks • Security of embedded devices, e.g., medical devices, cars • Trusted path	• Explain the concept of code integrity and code signing and the scope it applies to [Familiarity] • Discuss the concept of root of trust and the process of secure boot and secure loading [Familiarity] • Describe the mechanism of remote attestation of system integrity [Familiarity] • Summarize the goals and key primitives of TPM [Familiarity] • Identify the threats of plugging peripherals into a device [Familiarity] • Identify physical attacks and countermeasures [Familiarity] • Identify attacks on non-PC hardware platforms [Familiarity] • Discuss the concept and importance of trusted path [Familiarity]
Readings : [WL14]	

Unit 9: Digital Forensics (25)	
Competences Expected: a,g	
Topics	**Learning Outcomes**
• Basic Principles and methodologies for digital forensics • Design systems with forensic needs in mind • Rules of Evidence - general concepts and differences between jurisdictions and Chain of Custody • Search and Seizure of evidence: legal and procedural requirements • Digital Evidence methods and standards • Techniques and standards for Preservation of Data • Legal and Reporting Issues including working as an expert witness • OS/File System Forensics • Application Forensics • Web Forensics • Network Forensics • Mobile Device Forensics • Computer/network/system attacks • Attack detection and investigation • Anti-forensics	• Describe what is a Digital Investigation is, the sources of digital evidence, and the limitations of forensics [Familiarity] • Explain how to design software to support forensics [Familiarity] • Describe the legal requirements for use of seized data [Familiarity] • Describe the process of evidence seizure from the time when the requirement was identified to the disposition of the data [Familiarity] • Describe how data collection is accomplished and the proper storage of the original and forensics copy [Familiarity] • Conduct data collection on a hard drive [Usage] • Describe a person's responsibility and liability while testifying as a forensics examiner [Familiarity] • Recover data based on a given search term from an imaged system [Usage] • Reconstruct application history from application artifacts [Familiarity] • Reconstruct web browsing history from web artifacts [Familiarity] • Capture and interpret network traffic [Familiarity] • Discuss the challenges associated with mobile device forensics [Familiarity]
Readings : [WL14]	

Unit 10: Secure Software Engineering (25)	
Competences Expected: a,g,i,c	
Topics	**Learning Outcomes**
• Building security into the software development lifecycle • Secure design principles and patterns • Secure software specifications and requirements • Secure software development practices • Secure testing- the process of testing that security requirements are met (including static and dynamic analysis).	• Describe the requirements for integrating security into the SDL [Familiarity] • Apply the concepts of the Design Principles for Protection Mechanisms, the Principles for Software Security (Viega and McGraw), and the Principles for Secure Design (Morrie Gasser) on a software development project [Familiarity] • Develop specifications for a software development effort that fully specify functional requirements and identifies the expected execution paths [Familiarity]
Readings : [WL14]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[WL14] Stallings. W and Brown. L. *Computer Security: Principles and Practice*. Pearson Education, Limited, 2014. ISBN: 9780133773927.

Ministry of Education (MINEDU)
Sillabus 2021-I

1. COURSE

CS3P1. Parallel and Distributed Computing (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	4 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	• CS212. Analysis and Design of Algorithms. (5th Sem) • CS231. Networking and Communication. (5th Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

The last decade has brought explosive growth in computing with multiprocessors, including Multi-core processors and distributed data centers. As a result, computing parallel and distributed has become a widely elective subject to be one of the main components in the mesh studies in computer science undergraduate. Both parallel and distributed computing the simultaneous execution of multiple processes, whose operations have the potential to intercalar in a complex way. Parallel and distributed computing builds on foundations in many areas, including understanding the fundamental concepts of systems, such as: concurrency and parallel execution, consistency in state / memory manipulation, and latency. The communication and coordination between processes has its foundations in the passage of messages and models of shared memory of computing and algorithmic concepts like atomicity, consensus and conditional waiting.Achieving acceleration in practice requires an understanding of parallel algorithms, strategies for decomposition problem, systems architecture, implementation strategies and analysis of performance. Distributed systems highlight the problems of security and tolerance to Failures, emphasize the maintenance of the replicated state and introduce additional problems in the field of computer networks.

5. GOALS

- That the student is able to create parallel applications of medium complexity by efficiently leveraging machines with multiple cores.
- That the student is able to compare sequential and parallel applications.
- That the student is able to convert, when the situation warrants, sequential applications to parallel efficiently

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

c) An ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability. (**Usage**)

g) The broad education necessary to understand the impact of computing solutions in a global, economic, environmental, and societal context. (**Usage**)

a5) (**5**)

b2) (**2**)

7. TOPICS

Unit 1: Parallelism Fundamentals (18)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Multiple simultaneous computations • Goals of parallelism (e.g., throughput) versus concurrency (e.g., controlling access to shared resources) • Parallelism, communication, and coordination – Parallelism, communication, and coordination – Need for synchronization • Programming errors not found in sequential programming – Data races (simultaneous read/write or write/write of shared state) – Higher-level races (interleavings violating program intention, undesired non-determinism) – Lack of liveness/progress (deadlock, starvation)	• Distinguish using computational resources for a faster answer from managing efficient access to a shared resource [Familiarity] • Distinguish multiple sufficient programming constructs for synchronization that may be inter-implementable but have complementary advantages [Familiarity] • Distinguish data races from higher level races [Familiarity]
Readings : [Pac11], [Mat14], [**quinnz**], [Geo10]	

Unit 2: Parallel Architecture (12)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Multicore processors • Shared vs distributed memory • Symmetric multiprocessing (SMP) • SIMD, vector processing • GPU, co-processing • Flynn's taxonomy • Instruction level support for parallel programming – Atomic instructions such as Compare and Set • Memory issues – Multiprocessor caches and cache coherence – Non-uniform memory access (NUMA) • Topologies – Interconnects – Clusters – Resource sharing (e.g., buses and interconnects)	• Explain the differences between shared and distributed memory [Assessment] • Describe the SMP architecture and note its key features [Assessment] • Characterize the kinds of tasks that are a natural match for SIMD machines [Usage] • Describe the advantages and limitations of GPUs vs CPUs [Usage] • Explain the features of each classification in Flynn's taxonomy [Usage] • Describe the challenges in maintaining cache coherence [Familiarity] • Describe the key performance challenges in different memory and distributed system topologies [Familiarity]
Readings : [Pac11], [KH13], [SK10], [Geo10]	

Unit 3: Parallel Decomposition (18)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Need for communication and coordination/synchronization • Independence and partitioning • Basic knowledge of parallel decomposition concept • Task-based decomposition – Implementation strategies such as threads • Data-parallel decomposition – Strategies such as SIMD and MapReduce • Actors and reactive processes (e.g., request handlers)	• Explain why synchronization is necessary in a specific parallel program [Usage] • Identify opportunities to partition a serial program into independent parallel modules [Familiarity] • Write a correct and scalable parallel algorithm [Usage] • Parallelize an algorithm by applying task-based decomposition [Usage] • Parallelize an algorithm by applying data-parallel decomposition [Usage] • Write a program using actors and/or reactive processes [Usage]
Readings : [Pac11], [Mat14], [Qui03], [Geo10]	

Unit 4: Communication and Coordination (18)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Shared Memory • Consistency, and its role in programming language guarantees for data-race-free programs • Message passing – Point-to-point versus multicast (or event-based) messages – Blocking versus non-blocking styles for sending and receiving messages – Message buffering (cross-reference PF/Fundamental Data Structures/Queues) • Atomicity – Specifying and testing atomicity and safety requirements – Granularity of atomic accesses and updates, and the use of constructs such as critical sections or transactions to describe them – Mutual Exclusion using locks, semaphores, monitors, or related constructs ∗ Potential for liveness failures and deadlock (causes, conditions, prevention) – Composition ∗ Composing larger granularity atomic actions using synchronization ∗ Transactions, including optimistic and conservative approaches • Consensus – (Cyclic) barriers, counters, or related constructs • Conditional actions – Conditional waiting (e.g., using condition variables)	• Use mutual exclusion to avoid a given race condition [Usage] • Give an example of an ordering of accesses among concurrent activities (eg, program with a data race) that is not sequentially consistent [Familiarity] • Give an example of a scenario in which blocking message sends can deadlock [Usage] • Explain when and why multicast or event-based messaging can be preferable to alternatives [Familiarity] • Write a program that correctly terminates when all of a set of concurrent tasks have completed [Usage] • Give an example of a scenario in which an attempted optimistic update may never complete [Familiarity] • Use semaphores or condition variables to block threads until a necessary precondition holds [Usage]
Readings : [Pac11], [Mat14], [Qui03], [Geo10]	

Unit 5: Parallel Algorithms, Analysis, and Programming (18)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Critical paths, work and span, and the relation to Amdahl's law • Speed-up and scalability • Naturally (embarrassingly) parallel algorithms • Parallel algorithmic patterns (divide-and-conquer, map and reduce, master-workers, others) – Specific algorithms (e.g., parallel MergeSort) • Parallel graph algorithms (e.g., parallel shortest path, parallel spanning tree) (cross-reference AL/Algorithmic Strategies/Divide-and-conquer) • Parallel matrix computations • Producer-consumer and pipelined algorithms • Examples of non-scalable parallel algorithms	• Define "critical path", "work", and "span" [Familiarity] • Compute the work and span, and determine the critical path with respect to a parallel execution diagram [Usage] • Define "speed-up" and explain the notion of an algorithm's scalability in this regard [Familiarity] • Identify independent tasks in a program that may be parallelized [Usage] • Characterize features of a workload that allow or prevent it from being naturally parallelized [Familiarity] • Implement a parallel divide-and-conquer (and/or graph algorithm) and empirically measure its performance relative to its sequential analog [Usage] • Decompose a problem (eg, counting the number of occurrences of some word in a document) via map and reduce operations [Usage] • Provide an example of a problem that fits the producer-consumer paradigm [Usage] • Give examples of problems where pipelining would be an effective means of parallelization [Usage] • Implement a parallel matrix algorithm [Usage] • Identify issues that arise in producer-consumer algorithms and mechanisms that may be used for addressing them [Usage]
Readings : [Mat14], [Qui03], [Geo10]	

Unit 6: Parallel Performance (18)	
Competences Expected: a,b,c	
Topics	**Learning Outcomes**
• Load balancing • Performance measurement • Scheduling and contention (cross-reference OS/Scheduling and Dispatch) • Evaluating communication overhead • Data management – Non-uniform communication costs due to proximity (cross-reference SF/Proximity) – Cache effects (e.g., false sharing) – Maintaining spatial locality • Power usage and management	• Detect and correct a load imbalance [Usage] • Calculate the implications of Amdahl's law for a particular parallel algorithm (cross-reference SF/Evaluation for Amdahl's Law) [Usage] • Describe how data distribution/layout can affect an algorithm's communication costs [Familiarity] • Detect and correct an instance of false sharing [Usage] • Explain the impact of scheduling on parallel performance [Familiarity] • Explain performance impacts of data locality [Familiarity] • Explain the impact and trade-off related to power usage on parallel performance [Familiarity]
Readings : [Pac11], [Mat14], [KH13], [SK10], [Geo10]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[Geo10] Gerhard Wellein Georg Hager. *Introduction to High Performance Computing for Scientists and Engineers (Chapman & HallCRC Computational Science)*. Ed. by CRC Press. 1st. 2010. ISBN: 978-1439811924.

[KH13] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. 2nd. Morgan Kaufmann, 2013. ISBN: 978-0-12-415992-1.

[Mat14] Norm Matloff. *Programming on Parallel Machines*. University of California, Davis, 2014. URL: `http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf`.

[Pac11] Peter S. Pacheco. *An Introduction to Parallel Programming*. 1st. Morgan Kaufmann, 2011. ISBN: 978-0-12-374260-5.

[Qui03] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. 1st. McGraw-Hill Education Group, 2003. ISBN: 0071232656.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. 1st. Addison-Wesley Professional, 2010. ISBN: 0131387685, 9780131387683.