

Universidad Católica San Pablo
Facultad de Ingeniería y Computación
Escuela Profesional de
Ciencia de la Computación
SILABO



CS343. Lenguajes de Programación (Obligatorio)

2017-II

1. DATOS GENERALES

1.1 CARRERA PROFESIONAL	:	Ciencia de la Computación
1.2 ASIGNATURA	:	CS343. Lenguajes de Programación
1.3 SEMESTRE ACADÉMICO	:	7 ^{mo} Semestre.
1.4 PREREQUISITO(S)	:	CS211T. Teoría de la Computación. (4 ^{to} Sem) , CS210T. Análisis y Diseño de Algoritmos. (5 ^{to} Sem)
1.5 CARÁCTER	:	Obligatorio
1.6 HORAS	:	2 HT; 2 HP; 2 HL;
1.7 CRÉDITOS	:	4

2. DOCENTE

Dra. Marcela Quispe Cruz

- Dr. Ciencia de la Computación, Pontificia Universidad Católica de Rio de Janeiro, Brasil, 2014.
- Mag. Ciencia de la Computación, Universidad de Pernambuco, Brasil, 2009.

3. FUNDAMENTACIÓN DEL CURSO

A pesar de que los algoritmos han sido diseñados y escritos por lo menos desde el tiempo de Euclides; es que tan sólo en los últimos cincuenta años (desde el desarrollo de la computadora digital) los métodos de expresar algoritmos han sido objeto de un extenso estudio. En la actualidad existen distintos paradigmas de programación, cientos de lenguajes de programación en uso activo, muchos más en existencia y aún más por ser diseñados. El propósito de este curso es el dar una introducción a los principios del estudio de la programación, y brindar los fundamentos básicos en este tópico. Al brindar un estudio exhaustivo de los principios del diseño de los lenguajes de programación es que este curso pretende convertir al estudiante en un mejor programador. Adicionalmente este curso es útil si se necesita tomar la decisión acerca de que lenguaje de programación usar para un proyecto, o si alguna vez necesita diseñar su propio lenguaje.

4. SUMILLA

1. El desarrollo histórico de los lenguajes de programación 2. Lenguajes Imperativos 3. Orientación a Objetos 4. Lenguajes Funcionales 5. Lenguajes Lógicos 6. Otros Paradigmas

5. OBJETIVO GENERAL

- Capacitar a los estudiantes para entender los lenguajes de programación desde diferentes tipos de vista, según el modelo subyacente, los componentes fundamentales presentes en todo lenguaje de programación y como objetos formales dotados de una estructura y un significado según diversos enfoques.

6. CONTRIBUCIÓN A LA FORMACIÓN PROFESIONAL Y FORMACIÓN GENERAL

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. [Nivel Bloom: 3]
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. [Nivel Bloom: 4]
- i) Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. [Nivel Bloom: 3]
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. [Nivel Bloom: 4]

7. CONTENIDOS

UNIDAD 1: El desarrollo histórico de los lenguajes de programación (4 horas)

Nivel Bloom: 2

OBJETIVO GENERAL

CONTENIDO

- Reconocer el desarrollo histórico de los lenguajes de programación
- Identificar los paradigmas que agrupan a la mayoría de lenguajes de programación existentes hoy en día
- Discutir entre los distintos paradigmas y establecer sus ventajas y desventajas
- Establecer otros enfoques para la clasificación de los lenguajes de programación
- Diferenciar entre los distintos enfoques estructurales, desde el análisis léxico hasta el semántico
- Identificar el concepto de abstracción entre los distintos paradigmas
- Diferenciar entre la compilación y la interpretación en la ejecución de programas
- Reconocer como funciona un programa a nivel de computador

- Historia de los Lenguajes de Programación
- Paradigmas existentes
- Estructura de un programa: Léxico, Sintáctico y Semántico
- BNF
- Técnicas para la reducción de la complejidad en los programas
- Procesamiento de programas: Interpretación vs Compilación

Lecturas: [Bal and Grune, 1994], [Sebesta, 2005], [Sethi, 1996]

UNIDAD 2: Lenguajes Imperativos (4 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Identificar los principios de la programación imperativa ▪ Determinar las bases del imperativismo: secuencialidad, selección y repetición ▪ Aprender cómo los lenguajes imperativos manejan datos y asigna valores ▪ Aprender el concepto de ortogonalidad en un lenguaje de programación ▪ Diseña e implementa un programa en un lenguaje de programación imperativo 	<ul style="list-style-type: none"> ▪ Introducción ▪ Manejo de datos y tipos ▪ Asignaciones y Expresiones ▪ Flujos de control ▪ Componentes de un programa imperativo ▪ Ejemplos de programas imperativos
Lecturas: [Bal and Grune, 1994], [Sebesta, 2005], [Sethi, 1996]	

UNIDAD 3: Orientación a Objetos (8 horas)	
Nivel Bloom: 4	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Identificar los principios básicos en los cuales se basa la programación orientada a objetos ▪ Analiza como pasar del dominio de un problema a un modelado orientado a objetos ▪ Aprende como representar a nivel de lenguaje y a nivel de abstracción un caso problema ▪ Aprende la sintaxis de un lenguaje de programación orientado a objetos puro ▪ Implementa un programa en lenguaje de programación orientado a objetos ▪ Analiza los distintos tipos de herencia que presentan los lenguajes de programación orientados a este paradigma y examina sus ventajas y desventajas 	<ul style="list-style-type: none"> ▪ Introducción a los principios de la programación orientada a objetos ▪ Conceptos básicos: Clases, Herencia y Polimorfismo ▪ Binding Dinámico ▪ Semántica referencial ▪ Ejemplos de programas orientados a objetos
Lecturas: [Bal and Grune, 1994], [Sebesta, 2005], [Sethi, 1996]	

UNIDAD 4: Lenguajes Funcionales (4 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Reconocer los principios fundamentales del paradigma funcional ▪ Comparar las ventajas de la orientación funcional sobre otros esquemas ▪ Analiza el concepto de funciones y lo aplica en la solución de problemas ▪ Establece la facilidad del uso de lenguajes funcionales para casos en estructuras de datos y recursividad ▪ Investiga sobre el cálculo lambda ▪ Diseña e implementa programas en algún tipo de lenguaje funcional 	<ul style="list-style-type: none"> ▪ Introducción a los lenguajes funcionales ▪ Definición de función ▪ Listas ▪ Tipos y Polimorfismo ▪ Funciones de orden superior ▪ Lazy Evaluation ▪ Ejemplos de programas funcionales
Lecturas: [Bal and Grune, 1994], [Sebesta, 2005], [Sethi, 1996]	

UNIDAD 5: Lenguajes Lógicos (4 horas)	
Nivel Bloom: 3	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Comprender el modo de operación de los lenguajes lógicos orientados al logro de metas ▪ Analizar el encadenamiento hacia adelante o hacia atrás ▪ Aprender un lenguaje orientado al paradigma lógico ▪ Diseñar e implementar programas en lenguajes de programación orientados a objetos 	<ul style="list-style-type: none"> ▪ Principios ▪ Cláusulas de Horn ▪ Variables Lógicas ▪ Relaciones y Estructuras de Datos ▪ Control del orden de búsqueda ▪ Ejemplos de programas basados en el paradigma lógico
Lecturas: [Bal and Grune, 1994], [Sebesta, 2005], [Sethi, 1996]	

UNIDAD 6: Otros Paradigmas (2 horas)	
Nivel Bloom: 2	
OBJETIVO GENERAL	CONTENIDO
<ul style="list-style-type: none"> ▪ Identifica otros paradigmas presentes en nuestro medio ▪ Analiza si es que los paradigmas estudiados son novedosos o solamente una consecuencia de los principales paradigmas analizados ▪ Crítica la máquina de Von Neumann en base a los conocimientos de su arquitectura ▪ Diseña e implementa programas en un lenguaje de programación basado en los paradigmas estudiados 	<ul style="list-style-type: none"> ▪ Programación Paralela ▪ Programación Distribuida ▪ Crítica a la máquina de Von Neumann
Lecturas: [Bal and Grune, 1994], [Mueller, 2004], [Sebesta, 2005], [Sethi, 1996]	

8. METODOLOGÍA

El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.

El profesor del curso presentará demostraciones para fundamentar clases teóricas.

El profesor y los alumnos realizarán prácticas

Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.

9. EVALUACIONES

Evaluación Permanente 1 : 20 %

Examen Parcial : 30 %

Evaluación Permanente 2 : 20 %

Examen Final : 30 %

Referencias

[Bal and Grune, 1994] Bal, H. E. and Grune, D. (1994). *Programming Language Essentials*. Addison-Wesley.

[Mueller, 2004] Mueller, C. (2004). *Addressing: The root of all programming evils*. Proceedings of the 28th Annual International Computer Software and Applications Conference.

[Sebesta, 2005] Sebesta, R. W. (2005). *Concepts of Programming Languages*. Addison-Wesley.

[Sethi, 1996] Sethi, R. (1996). *Programming Languages: Concepts and Constructs, Second Edition*. Addison Wesley Publishing Company.