UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Licenciatura en Ciencias de la Computación

Facultad de Ciencias

Programa de la asignatura

Denominación de la asignatura:

Lenguajes de Programación

Clave:	Semestre:	Eje temático:	No. Créditos:
	5	Integración Teoría-Práctica	10

Carácter: Obligatoria	Horas		Horas por semana	Total de Horas
Tipo: Teórico-Práctica	Teoría:	Práctica:	7	112
	3	4		
Modalidad: Curso	Duración del programa: Semestral			

Asignatura con seriación obligatoria antecedente: Álgebra Superior II; Gráficas y Juegos; Estructuras de Datos

Asignatura con seriación obligatoria subsecuente: Ninguna

Asignatura con seriación indicativa antecedente: Autómatas y Lenguajes Formales; Lógica Computacional

Asignatura con seriación indicativa subsecuente: Compiladores

Objetivo general:
Conocer y aplicar los principios y componentes en el diseño de los lenguajes de programación y contará con las herramientas básicas para analizar formalmente diversas de sus características.

Índice temático

Unidad	Temas	Horas	
		Teóricas	Prácticas
I	Introducción	3	4
II	Fundamentos	9	12
III	Paradigma funcional	12	16
IV	Paradigma imperativo	12	16
V	Paradigma orientado a objetos	12	16
	Total de horas:	**48**	**64**
	Suma total de horas:	**112**	

Contenido temático	
Unidad	**Tema**
I Introducción	
I.1	Historia de los lenguajes de programación.
I.2	Breve discusión de los paradigmas clásicos (imperativo, orientado a objetos, funcional, lógico).
I.3	Programación en pequeña y gran escala.
II Fundamentos	
II.1	Herramientas matemáticas: definiciones inductivas; inducción estructural; reglas de inferencia; sistemas de transición.
II.2	Sintaxis: niveles de sintaxis (concreta y abstracta); manejo de variables (ligado y alcance); análisis sintáctico (*parsing*).
II.3	Semántica: estilos de semántica (denotacional, operacional, axiomática); semántica estática y dinámica; sistemas de tipos.
II.4	Expresiones aritméticas y booleanas; expresiones `let`.
III Paradigma funcional	
III.1	Funciones anónimas, abstracción lambda y aplicación de funciones.
III.2	Orden y estrategias de evaluación; evaluación ansiosa (*eager*) y perezosa (*lazy*).
III.3	Tipos función (cálculo lambda con tipos simples).
III.4	Estilos de tipado: tipado dinámico vs. tipado estático.
III.5	Recursión: funciones con nombre, puntos fijos, definiciones con `letrec` o `fix`.
III.6	Tipos de datos finitos e infinitos: registros (productos), variantes (sumas), tipos recursivos.
III.7	Polimorfismo.
IV Paradigma imperativo	
IV.1	Máquinas abstractas.
IV.2	Procedimientos y bloques de programa.
IV.3	El enunciado de asignación.
IV.4	Registros de activación; manejo de memoria; recolección de basura.
IV.5	Mecanismos de paso de parámetros (por valor, por nombre, por referencia, por necesidad).
IV.6	Estructuras de control simples; iteradores.
IV.7	Manejo de excepciones.
IV.8	Continuaciones.
V Paradigma orientado a objetos	
V.1	Fundamentos: representaciones múltiples, encapsulamiento, subtipado, herencia, recursión abierta.
V.2	Polimorfismo de subtipos.
V.3	Modelado y representación de objetos.
V.4	Despacho dinámico y reemplazo de métodos (*overriding*).
V.5	Formalismos: Java Peso Pluma (*Featherweight Java*), SOOL.

Bibliografía básica:

1. Mitchell J., *Concepts in Programming Languages*. Cambridge University Press 2003.
2. Pierce B.C., *Types and Programming Languages*. MIT Press 2002.

Bibliografía complementaria:

1. Harper R., *Practical Foundations for Programming Languages*. Working draft, 2010. Recuperado el día 11 de febrero de 2010 de la página http://www.cs.cmu.edu/~rwh/plbook/book.pdf
2. Mitchell J., *Foundations for Programming Languages*. MIT Press 1996.
3. Friedman D. P., Wand M., *Essentials of Programming Languages. 3rd edition*. MIT Press, 2008.
4. Krishnamurthi S., *Programming Languages Application and Interpretation*; Version 26.04.2007. Disponible en línea en http://www.cs.brown.edu/~sk/Publications/Books/ProgLangs/

Sugerencias didácticas:

Exposición oral	**(X)**
Exposición audiovisual	**( )**
Ejercicios dentro de clase	**(X)**
Ejercicios fuera del aula	**(X)**
Seminarios	**( )**
Lecturas obligatorias	**(X)**
Trabajo de investigación	**( )**
Prácticas de taller o laboratorio	**(X)**
Prácticas de campo	**( )**

Otras: Se sugiere complementar la teoría con prácticas de laboratorio orientadas hacia la implementación incremental de un intérprete para un prototipo sencillo de lenguaje de programación utilizando para tal propósito un lenguaje funcional.

Métodos de evaluación:

Exámenes parciales	**(X)**
Examen final escrito	**( )**
Trabajos y tareas fuera del aula	**(X)**
Exposición de seminarios por los alumnos	**( )**
Participación en clase	**(X)**
Asistencia	**( )**
Seminario	**( )**

Otras: Prácticas de laboratorio en un lenguaje funcional.

Perfil profesiográfico:

Egresado preferentemente de la Licenciatura en Ciencias de la Computación o Matemático con especialidad en Computación. Es conveniente que posea un posgrado en la disciplina. Con experiencia docente.