



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 II

Clave:	Semestre: 7-8	Eje temático: Lenguajes de Programación			No. Créditos: 10
Carácter: Optativa		Horas		Horas por semana	Total de Horas
Tipo: Téorico-Práctica		Teoría:	Práctica:	7	112
		3	4		
Modalidad: Curso		Duración del programa: Semestral			

Asignatura con seriación indicativa antecedente: Lenguajes de Programación

Asignatura con seriación indicativa subsecuente: Ninguna

Objetivo general:

Estudiar a profundidad diversos conceptos avanzados subyacentes al diseño de lenguajes de programación, así como técnicas formales que ayuden a la implementación de éstos.

Conocer a profundidad diversos principios y conceptos avanzados que resultan ser parte fundamental de algunos componentes, en el diseño de los lenguajes de programación. De esta manera se comprenderán las características de dichos conceptos tanto de manera teórica como en la práctica.

Índice temático

Unidad	Temas	Horas	
		Teóricas	Prácticas
I	Introducción	1.5	2
II	Polimorfismo	6	8
III	Tipos de Datos Abstractos	6	8
IV	Tipos Recursivos	7.5	10
V	Tipado Dinámico	6	8
VI	Inferencia de Tipos	6	8
VII	Recolección de Basura	7.5	10
VIII	Continuaciones	7.5	10
Total de horas:		48	64
Suma total de horas:		112	

Contenido temático	
Unidad	Tema
I Introducción	
I.1	Revisión de conceptos y aspectos de lenguajes de programación estudiados en el curso obligatorio de lenguajes de programación.
II Polimorfismo	
II.1	Clases de polimorfismo.
II.2	Polimorfismo paramétrico y ad-hoc.
II.3	Funciones parametrizadas.
II.4	Interacción, evaluación, semántica y efectos del uso del polimorfismo.
II.5	Polimorfismo de subtipos (subtipado algorítmico).
III Tipos de Datos Abstractos	
III.1	Tipos de datos abstractos.
III.2	Tipos existenciales y su semántica.
III.3	Implementación de tipos de datos abstractos.
III.4	Módulos.
IV Tipos Recursivos	
IV.1	Definición y conceptos relacionados con recursión.
IV.2	Declaración de tipos recursivos (tipos isorrecursivos y equirrecursivos).
IV.3	Funciones que involucran tipos recursivos.
IV.4	Semántica de tipos recursivos.
IV.5	Tipos inductivos.
IV.6	Tipos perezosos coinductivos (expresividad y ventajas de la evaluación perezosa).
V Tipado Dinámico	
V.1	El cálculo lambda sin tipos, panorama de lenguajes sin tipos.
V.2	Tipado dinámico.
V.3	Tipado híbrido.
VI Inferencia de Tipos	
VI.1	Introducción a la inferencia de tipos.
VI.2	El sistema de tipos de Hindley-Milner.
VI.3	Restricciones de tipos.
VI.4	Algoritmos de unificación y de inferencia de tipos.
VI.5	Semántica y coherencia de las expresiones.
VII Recolección de Basura	
VII.1	Introducción a la recolección de basura (objetos vivos, basura, apuntadores colgantes, por qué la recolección de basura).
VII.2	Algoritmos clásicos de recolección de basura (conteo de referencias, marcado y barrido, marcado y compactación, detente y copia).
VII.3	Técnicas de recolección de basura (recolección generacional e incremental)
VIII Continuaciones	
VIII.1	Introducción a las continuaciones.
VIII.2	Continuation Passing Style (CPS).
VIII.3	Semántica de las continuaciones.

VIII.4	Optimizaciones en continuaciones.
VIII.5	Implementación de continuaciones (Máquinas abstractas).
VIII.6	Corrutinas.

Bibliografía básica:

1. Mitchell J., *Concepts in Programming Languages*, Cambridge University Press, 2003.
2. Pierce, B.C., *Types and Programming Languages*, MIT Press, 2002.
3. Felleisen, Matthias; Findler, Robert Bruce y Flatt, Matthew, *Semantics Engineering with PLT Redex*, The MIT Press; agosto 2009.

Bibliografía complementaria:

1. Harper R., *Practical Foundations for Programming Languages*, Working Draft 2010. Recuperado el 11 de febrero de 2010 del URL <http://www.cs.cmu.edu/~rwh/plbook/book.pdf>
2. Friedman D.P., Wand M. *Essentials of Programming Languages*, 3a edición, MIT Press, 2008.
3. Krishnamurthi S., *Programming Languages Application and Interpretation*, Versión 04, 2007. Disponible en línea en <http://www.cs.brown.edu/~sk/Publications/Books/PorgLangs>
4. Mitchel J., *Foundations for Programming Languages*, MIT Press, 1996.
5. Appel, Andrew, *Compiling with Continuations*, Cambridge University Press, 1992.
6. Miller, Frederic P., Vandome, Agnes F., McBrewster, John., *Control flow: Computer Science, Subroutine, Imperative programming, Functional programming, Computer Program, Programming Language, Coroutine, Continuation, Signal Computing*, Alphascript Publishing, 2009.
7. Jones, Richard, Lins, Rafael D., *Garbage Collection: Algorithms for Automatic Dynamic Memory Management*, Wiley, 1996.
8. Jones, Mark P., *Qualified Types: Theory and Practice*, Cambridge University Press, 2003.

Sugerencias didácticas:		Métodos de evaluación:	
Exposición oral	()	Exámenes parciales	(X)
Exposición audiovisual	()	Examen final escrito	()
Ejercicios dentro de clase	(X)	Trabajos y tareas fuera del aula	(X)
Ejercicios fuera del aula	()	Prácticas de laboratorio	()
Seminarios	(X)	Exposición de seminarios por los alumnos	()
Lecturas obligatorias	(X)	Participación en clase	(X)
Trabajo de investigación	(X)	Asistencia	()
Prácticas de taller o laboratorio	(X)	Proyectos de programación	(X)
Prácticas de campo	()	Proyecto final	()
		Seminario	()
Otras: _____		Otras: _____	

Perfil profesiográfico:

Matemático, físico, actuario o Licenciado en Ciencias de la Computación, especialista en el área de la asignatura a juicio del comité de asignación de cursos. Con experiencia docente.