

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

Programa de la asignatura: **PROGRAMACIÓN**

LICENCIATURA: INFORMÁTICA		CLAVE: 1369		
SEMESTRE: 3°	PLAN: 2005	ÁREA: Informática (Desarrollo de sistemas)		
CARÁCTER: Obligatoria		Hora / Semana / Semestre		Créditos:
		Teóricas	Prácticas	8
		4	0	
MODALIDAD: Laboratorio				
SERIACIÓN ANTECEDENTE OBLIGATORIA: Introducción a la programación, 1er. Semestre				
SERIACIÓN SUBSECUENTE OBLIGATORIA: Ninguna				

**OBJETIVO:**  
AL FINALIZAR EL CURSO EL ALUMNO CONOCERÁ LA EVOLUCIÓN DE LOS LENGUAJES DE PROGRAMACIÓN, ASÍ COMO LAS DIFERENTES FILOSOFÍAS (PARADIGMAS) QUE EMPLEAN PARA DESCRIBIR MODELOS DE LA REALIDAD.

TEMAS	Número de horas
I. DEFINICIÓN DE UN LENGUAJE DE PROGRAMACIÓN	12
II. PARADIGMA IMPERATIVO	12
III. PARADIGMA ORIENTADO A OBJETOS	20
IV. PARADIGMA FUNCIONAL	10
V. PARADIGMA LÓGICO	10
	<b>TOTAL 64 HORAS</b>

**I. DEFINICIÓN DE UN LENGUAJE DE PROGRAMACIÓN**

12

Objetivo: Al término de esta unidad, el alumno definirá qué es un lenguaje de programación; podrá explicar las clasificaciones que se hacen de éstos, sus niveles de abstracción, los paradigmas que los implementan, su evolución e historia, su definición formal (a través de la sintaxis y semántica) y el proceso de traducción

1. ¿Qué es un lenguaje de programación?
2. Clasificaciones de los lenguajes de programación
3. Abstracciones en los lenguajes de programación
  - 3.1. De datos
  - 3.2. De control
4. Historia y evolución de los lenguajes de programación
5. Definición de un lenguaje de programación
  - 5.1. Sintaxis
    - 5.1.1. Léxico
    - 5.1.2. Metalenguajes y notación BNF
  - 5.2. Semántica
6. Traducción de los lenguajes de programación
  - 6.1. Compilación
    - 6.1.1. Programa fuente
    - 6.1.2. Análisis léxico
    - 6.1.3. Análisis sintáctico
    - 6.1.4. Análisis semántico
    - 6.1.5. Generación de código intermedio
    - 6.1.6. Programa objeto
    - 6.1.7. Optimizador de código
    - 6.1.8. Tabla de símbolos
  - 6.2. Interpretación

**II. PARADIGMA IMPERATIVO**

12

Objetivo: Al término de esta unidad, el alumno explicará los conceptos más importantes que dan soporte y fundamento al paradigma de programación imperativa (celda de memoria variable, operaciones de asignación y repetición y transformación de datos) y los aplicará en la resolución de problemas algorítmicos propios de la automatización de la información y manipulación de datos.

1. Definición
2. Modularidad
3. Concepto de celda de memoria variable
4. Operaciones de asignación
5. Operaciones de repetición
6. Secuencia de transformación de datos
7. Campos de aplicación

### **III. PARADIGMA ORIENTADO A OBJETOS**

20

Objetivo: Al término de esta unidad, el alumno explicará los conceptos más importantes que dan soporte y fundamento al paradigma de programación orientado a objetos (clase, objeto, herencia, polimorfismo y encapsulamiento) y los aplicará en la resolución de problemas algorítmicos propios de la automatización de la información y manipulación de datos

1. Definición
2. Clases y objetos
3. Atributos y métodos
4. Relaciones estáticas y dinámicas entre objetos
5. Herencia
6. Polimorfismo
  - 6.1. Sobrecarga
  - 6.2. Sobreescritura
7. Encapsulamiento (ocultamiento de la información)
8. Diagrama de clases
9. Campos de aplicación

### **IV. PARADIGMA FUNCIONAL**

10

Objetivo: Al término de esta unidad, el alumno explicará los conceptos más importantes que dan soporte y fundamento al paradigma de programación funcional (funciones, recursividad y listas) y los podrá aplicar en la resolución de problemas algorítmicos propios de la automatización de la información y manipulación de datos

1. Definición
2. Programas con funciones
3. Recursividad
4. Listas
5. Implementación de algoritmos
6. Lenguajes funcionales puros e híbridos
7. Introducción a los lenguajes funcionales
8. Campos de aplicación

### **V. PARADIGMA LÓGICO**

10

Objetivo: Al término de esta unidad, el alumno explicará los conceptos más importantes que dan soporte y fundamento al paradigma de programación lógico (hechos, reglas, cláusulas de Horn y predicados) y los podrá aplicar en la resolución de problemas algorítmicos propios de la automatización de la información y manipulación de datos

1. Definición
2. Hechos
3. Reglas
4. Cláusulas de Horn
5. Predicados
6. Introducción a los lenguajes lógicos
7. Campos de aplicación

**TOTAL: 64  
HORAS**

## BIBLIOGRAFÍA BÁSICA

1. ALFONSECA, Manuel et. al., *Compiladores e intérpretes: teoría y práctica*, Madrid, Pearson Educación, 2006.
2. BERLAGE. T., *Concepts and programming*, Inglaterra, Addison-Wesley, 1991
3. BOBADILLA, Jesús, *HTML Dinámico, ASP y JAVASCRIPT*, España, Alfaomega-Rama, 2001
4. BRONSON Gary J., *C++ para Ingeniería y ciencias*, México, Thomson-Learning, 1999, 884 p.p.
5. DECKER Rick y Stuart Hirsfield, *Máquina analítica*, México, Thomson-Learning, 2001.
6. CEBALLOS, Francisco Javier, *Microsoft Visual C++6 aplicaciones para Win32*, España, Alfaomega-Rama, 2002.
7. FARRET, *Introducción a la programación. Lógica y diseño*, México, Thomson Learning, 2002, (4ª. Ed.)
8. GRAHAM, Ian, *Métodos orientados a objetos*, México, Addison-Wesley, 1996.
9. KNUTH, Donald, *The art of Computer Programming*, Vol. 1, Fundamental Algorithms, EEUU, Addison-Wesley. , (2ª. Ed), 1997
10. LÓPEZ ROMÁN Leobardo, *Programación estructurada. Un enfoque algorítmico*. México, Alfaomega, 2002.
11. LÓPEZ ROMÁN Leobardo, *Programación estructurada en TurboPascal 7*, México, Alfaomega, 2002.
12. LOUDEN, Kenneth C., *Programming Languages: Principles and Practice*, Second Edition, Brooks Cole, 2003.
13. ORÓS, Juan Carlos, *Diseño de páginas Web interactivas con JavaScript*, Alfaomega-Rama, 2001, (2ª. Ed.)
14. PRATT, T.W., *Lenguajes de programación*, México, Prentice-Hall, 1998.
15. SEBESTA, Robert, *Concepts of Programming Lenguajes*, EEUU, Addison-Wesley, 2006, (7ª. Ed.)
16. SETHI, Ravi, *Lenguajes de programación*, México, Addison-Wesley, 1992.
17. SMITH, *C++ Programación Orientada a objetos*, México, Thomson Learning, 2002
18. WANG Paul, *Java con programación orientada a objetos y aplicaciones en la WWW*, Thomson Learning, México, 2002

## BIBLIOGRAFÍA COMPLEMENTARIA

1. BUDD, Timothy, *Introducción a la programación orientada a objetos*, México, Addison-Wesley, 1994.
2. CEBALLOS, Francisco Javier, *Enciclopedia de Microsoft Visual Basic 6*, España, Alfaomega-Rama, 2002.
3. CRAWLEY, Winston, *Programación estructurada*, México, Prentice-Hall, 1990.
4. TUCKER, Allen, *Lenguajes de programación*, México, McGraw-Hill, 1987, (2ª. ed.)
5. UREÑA, Luis A., et. al., *Fundamentos de Informática*, España, Alfaomega-Rama, 2002.
6. WATSON, Des, *High Level Lenguajes and their Compilers*, EEUU, Addison-Wesley, 1992.

**SUGERENCIAS DIDÁCTICAS:**

Exposición audiovisual	( X )
Exposición oral	( X )
Ejercicios dentro de la clase	( X )
Seminarios	( )
Lecturas obligatorias	( X )
Trabajos de investigación	( X )
Prácticas de taller o laboratorio	( X )
Prácticas de campo	( )

**SUGERENCIAS PARA LA EVALUACIÓN:**

Exámenes parciales	( X )
Exámenes finales	( X )
Trabajos y tareas fuera de aula	( X )
Participación en clase	( X )
Asistencia a prácticas	( X )
Otras	( )

**PERFIL PROFESIOGRÁFICO DEL DOCENTE:**

**ESTUDIOS REQUERIDOS**

Licenciatura en Informática o carrera afín, preferentemente con estudios de posgrado.

**EXPERIENCIA PROFESIONAL DESEABLE**

Experiencia mínima de 2 años en empresas relacionadas con el área o su equivalente.

Tener experiencia docente mínima de 2 años.

**OTROS REQUERIMIENTOS**

Haber cursado los módulos de didáctica y docencia que imparte la Facultad, para profesores de nuevo ingreso, así como cubrir satisfactoriamente los requisitos impuestos por el departamento de selección y reclutamiento de la Facultad de Contaduría y Administración.