



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA



PROGRAMA DE ESTUDIO

MODELOS DE PROGRAMACIÓN
ORIENTADA A OBJETOS

1332

3

6

Asignatura

Clave

Semestre

Créditos

INGENIERÍA ELÉCTRICA

INGENIERÍA
EN COMPUTACIÓN

INGENIERÍA ELÉCTRICA
Y ELECTRÓNICA

División

Departamento

Licenciatura

Asignatura:

Obligatoria ☒

Optativa ☐

Horas/semana:

Teóricas

Prácticas

Total

Horas/semestre:

Teóricas

Prácticas

Total

Modalidad: Curso teórico-práctico

Seriación obligatoria antecedente: Estructura de Datos y Algoritmos I

Seriación obligatoria consecuente: Ninguna

Objetivo(s) del curso:

El alumno aplicará programas con el paradigma orientado a objetos clases, así como el diseño de abstracciones que apoyan el diseño de software y bibliotecas reusables, empleando un enfoque de pruebas sistemático.

Temario

NÚM.	NOMBRE	HORAS
1.	Tipos, expresiones y control de flujo	14.0
2.	Herencia y polimorfismo	10.0
3.	Manejo de excepciones y errores	8.0
		<hr/>
		32.0
	Actividades prácticas	32.0
		<hr/>
	Total	64.0

1 Tipos, expresiones y control de flujo

Objetivo: El alumno aplicará las técnicas y herramientas de la programación orientada a objetos para la solución de problemas.

Contenido:

1.1 Generalidades.

- 1.1.1 Identificadores.
- 1.1.2 Palabras reservadas.
- 1.1.3 Comentarios.
- 1.1.4 Descripción de una clase.
- 1.1.5 Descripción de un objeto.

1.2 Tipos de datos.

- 1.2.1 Primitivos y su jerarquía.
- 1.2.2 Referencias o instancias.
- 1.2.3 Conversiones entre tipos primitivos (moldeado o casting).
- 1.2.4 Operadores aritméticos.
- 1.2.5 Operadores de asignación.
- 1.2.6 Operadores relacionales.
- 1.2.7 Operadores especiales (in/decremento (post o pre), concatenación, acceso a variables y métodos y de agrupación).
- 1.2.8 Operadores a nivel de bits.
- 1.2.9 Operadores lógicos.

1.3 Arreglos.

1.4 Tipos y ámbito de las variables.

- 1.4.1 Elementos estáticos.
- 1.4.2 Elementos constantes.

1.5 Tipos de clases (públicas, sin modificador, abstractas, finales e internas).

1.6 Estructuras de selección.

- 1.6.1 Estructura if-else.
- 1.6.2 Estructura switch-case.
- 1.6.3 Estructura ternaria.

1.7 Estructuras de selección.

- 1.7.1 Estructura while.
- 1.7.2 Estructura do-while.
- 1.7.3 Estructura for.

1.8 Interfaces (abstracción).

2 Herencia y polimorfismo

Objetivo: El alumno aplicará las diferentes propiedades de la programación orientada a objetos para la resolución de problemas.

Contenido:

- 2.1 Herencia.
- 2.2 Método constructor.

2.3 Polimorfismo (moldeado o casting entre tipos referencia o instancias).

2.4 Referencias a this y a la clase base.

2.5 Modificadores de acceso (encapsulamiento).

2.6 Paquetes.

3 Manejo de excepciones y errores

Objetivo: El alumno aplicará los diferentes tipos de errores y excepciones para generar programas y aplicaciones con calidad.

Contenido:

3.1 Jerarquía de clases de errores.

3.2 Definición y diferencia entre error y excepción.

3.3 Estructura try-catch-finally.

Bibliografía básica

Temas para los que se recomienda:

DEITEL, Paul, DEITEL, Harvey

C# 2010 for Programmers

Todos

4th edition

Prentice Hall, 2010

DEITEL, Paul, DEITEL, Harvey

C++ How to Program

Todos

8th edition

Prentice Hall, 2011

DEITEL, Paul, DEITEL, Harvey

Java How to Program (early objects) plus MyProgrammingLab

Todos

with Pearson eText 9th edition

Prentice Hall, 2011

SARANG, Poornachandras

Java Programming (Oracle Press)

Todos

McGraw-Hill Osborne Media, 2012

Bibliografía complementaria

Temas para los que se recomienda:

FLANAGAN, David

Java In A Nutshell

Todos

5th edition

O Reilly Media, 2005

Sugerencias didácticas

Exposición oral	<input checked="" type="checkbox"/>
Exposición audiovisual	<input checked="" type="checkbox"/>
Ejercicios dentro de clase	<input checked="" type="checkbox"/>
Ejercicios fuera del aula	<input checked="" type="checkbox"/>
Seminarios	<input type="checkbox"/>
Uso de software especializado	<input type="checkbox"/>
Uso de plataformas educativas	<input type="checkbox"/>

Lecturas obligatorias	<input checked="" type="checkbox"/>
Trabajos de investigación	<input checked="" type="checkbox"/>
Prácticas de taller o laboratorio	<input checked="" type="checkbox"/>
Prácticas de campo	<input type="checkbox"/>
Búsqueda especializada en internet	<input type="checkbox"/>
Uso de redes sociales con fines académicos	<input type="checkbox"/>

Forma de evaluar

Exámenes parciales	<input checked="" type="checkbox"/>
Exámenes finales	<input checked="" type="checkbox"/>
Trabajos y tareas fuera del aula	<input checked="" type="checkbox"/>

Participación en clase	<input checked="" type="checkbox"/>
Asistencia a prácticas	<input checked="" type="checkbox"/>

Perfil profesiográfico de quienes pueden impartir la asignatura

El profesor será egresado de la carrera de Ingeniería en Computación o licenciatura afín, con conocimientos y experiencia en el diseño de algoritmos y programas en el paradigma orientados a objetos.