

# FACULTAD DE INGENIERÍA

## ÁREA DE COMPUTACIÓN E INFORMÁTICA



**Nombre de la materia:**  
PROGRAMACIÓN ORIENTADA A OBJETOS  
**Clave de la materia:** 2223  
**Clave Facultad:**  
**Clave U.A.S.L.P.:** Clave CACEI: CI  
**Nivel del Plan de Estudios:** IC, II: 4 **No. de créditos:** 8  
**Horas/Clase/Semana:** 3 **Horas totales/Semestre:** 80  
**Horas/Práctica (y/o Laboratorio):** 2  
**Prácticas complementarias:**  
**Trabajo extra-clase Horas/Semana:** 3  
**Carrera/Tipo de materia:** Común del Área/Obligatoria  
**No. de créditos aprobados:**  
**Fecha última de Revisión Curricular:** 12/ 03/2010  
**Materia y clave de la materia requisito:**  
ESTRUCTURAS DE DATOS Y ALGORITMOS B 2222

### PROPÓSITO DEL CURSO

Existe una diversa cantidad de paradigmas o modelos de programación de computadoras. El paradigma imperativo se inicia con la programación no estructurada, posteriormente la programación estructurada y finalmente a partir de la década de los

ochentas y con mayor fuerza en los noventa se desarrolla el paradigma orientado a objetos. Este paradigma por ser el más alto nivel de abstracción en programación, en la actualidad, es indispensable para desarrollo de software.

### OBJETIVO DEL CURSO

Conocer y aplicar los conceptos, mecanismos y técnicas de la programación orientada a objetos mediante diversos

lenguajes de programación: Smalltalk, Objective-C, C++, C# y Java.

### CONTENIDO TEMÁTICO

#### PARTE I. PROGRAMACION ORIENTADA A OBJETOS EN SMALLTALK

Tiempo estimado: 25 hrs.

Objetivo:

Conocer y aplicar los conceptos de la programación orientada a objetos utilizando el lenguaje de programación Smalltalk.

1. Antecedentes históricos.
2. Objetos.
  - o Números enteros, fracciones, flotantes, caracteres, cadenas, arreglos, símbolos, lógicos y nulos.
3. Mensajes.
  - o Componentes y prioridad.
  - o Condicionales y de repetición.
4. Clases y métodos.
  - o Creación de nuevas clases y métodos.
  - o Variables de instancia, de clase, temporales, globales y workspace.
5. Composición, herencia y polimorfismo.
6. Pilas y colas con arreglos y listas.
7. Iteradores y colecciones.

#### PARTE II. PROGRAMACION ORIENTADA A OBJETOS EN OBJECTIVE-C Y EN C++

##### 1. Objective-C

Tiempo estimado: 10 hrs.

Objetivo:

Conocer y aplicar los conceptos de la programación orientada a objetos utilizando el lenguaje de programación Objective-C. Comparar las características que no incluye el lenguaje Smalltalk.

- a. Antecedentes históricos.
- b. Objetos, mensajes, clases y métodos.
- c. Herencia y polimorfismo.
- d. Estructuras de datos: pilas y colas con arreglos y listas.

##### 2. C++ (15 HRS)

Tiempo estimado: 15 hrs.

Objetivo:

Conocer y aplicar los conceptos de la programación orientada a objetos utilizando el lenguaje de programación C++. Comparar las

características que no incluyen los lenguajes Smalltalk, Objective-C.

- a. Antecedentes históricos.
- b. Objetos, mensajes, clases y métodos.
- c. Herencia simple y múltiple.
- d. Polimorfismo y funciones virtuales.
- e. Sobrecarga de operadores.
- f. Funciones y clases amigas.
- g. Flujos y archivos.
- h. Genericidad (templates) y STL.

### PARTE III. PROGRAMACION ORIENTADA A OBJETOS EN C# Y JAVA

#### 1. Java

Tiempo estimado: 20 hrs.

Objetivo:

Conocer y aplicar los conceptos de la programación orientada a objetos utilizando el lenguaje de programación Java. . Comparar las características que no incluyen los lenguajes Smalltalk, Objective-C, C++.

- a. Antecedentes históricos.

- b. Objetos, mensajes, clases y métodos.
- c. Herencia y polimorfismo.
- d. Interfaces.
- e. Colecciones.
- f. Excepciones.
- g. Multitarea.
- h. Applets y swing.

#### 2. C#

Tiempo estimado: 10 hrs.

Objetivo:

Conocer y aplicar los conceptos de la programación orientada a objetos utilizando el lenguaje de programación C#. Comparar las características que no incluyen los lenguajes Smalltalk, Objective-C, C++ y Java.

- a. Antecedentes históricos.
- b. Objetos, mensajes, clases y métodos.
- c. Campos y propiedades.
- d. Herencia y polimorfismo.
- e. Delegados y eventos.
- f. Formularios.

## METODOLOGÍA

El profesor imparte las clases con medios didácticos, desarrolla y presenta programas de ejemplo en clase y los estudiantes aportan con su participación. Se realizan ejercicios de programación en clase y como tarea,

dependiendo de los ejercicios estos pueden ser en forma individual o grupal. Durante el semestre se desarrolla un proyecto en Java mediante la plataforma de Greenfoot.

## EVALUACIÓN

Los exámenes representan el 100% de la evaluación. Se realizarán tres exámenes parciales en las fechas establecidas por la facultad, de acuerdo al Reglamento de Exámenes. Es requisito para presentar los exámenes haber cumplido con todas las tareas asignadas. Para presentar el tercer examen parcial, el examen

extraordinario, el examen a título y regularización; es requisito hacer entrega del sistema desarrollado (proyecto) en la materia de acuerdo con los lineamientos establecidos en el Sistema de Proyectos de Desarrollo Tecnológico Integrador del área.

## BIBLIOGRAFÍA

### Bibliografía básica

Programación en C, C++, Java y UML  
Luis Joyanes Aguilar, I. Zahonero M.  
Mc GrawHill, 2010

### Bibliografía Complementaria

Descubre Smalltalk  
Lalonde, W.  
Ediciones Díaz de Santos, 1997

Apuntes: Guía de enseñanza aprendizaje de tecnología orientada a objetos  
Héctor Gerardo Pérez González  
Área de Computación e Informática, UASLP, 1998

Programación orientada a objetos: un enfoque evolutivo  
Brad J. Cox  
Addison Wesley 1993