

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS
LICENCIATURA EN SISTEMAS COMPUTACIONALES**

Área de formación: Disciplinaria
Unidad académica: Programación Orientada a Objetos
Ubicación: Cuarto Semestre
Clave: 2087
Horas semana-mes: 4
Horas teoría: 2
Horas práctica: 2
Unidades CONAIC: 42.67
Prerrequisitos: Ninguno
Horas de infraestructura: 2
Créditos: 6

PRESENTACIÓN

Aun cuando la Programación Orientada a Objetos (POO) data de mediados de la década de los setenta con el lenguaje SmallTalk, no es hasta finales de la década de los noventa cuando toma mayor impulso en otros lenguajes de programación.

Actualmente diversos lenguajes de programación emigran hacia la POO y los lenguajes nuevos se crean bajo ese paradigma. Esto implica que los sistemas de información actuales tienen en su programación esta filosofía. Es por lo anterior indispensable que el estudiante conozca el paradigma de la POO.

OBJETIVO GENERAL

El alumno aplicará tópicos de programación orientada a objetos en un lenguaje de programación, que sirvan de base para el desarrollo de software.

UNIDAD I.- CONCEPTOS BASICOS DEL MODELO ORIENTADO A OBJETOS

TIEMPO APROXIMADO: 4 Horas

OBJETIVO DE LA UNIDAD: Conocer los conceptos básicos del enfoque orientado a objetos.

CONTENIDO

- 1.1 Reconocimiento de objetos y clases del mundo real y su interacción
- 1.2 La abstracción y el encapsulamiento como un proceso natural
- 1.3 La POO y la complejidad del software
- 1.4 Conceptos del ciclo de vida del software
 - 1.4.1 Especificaciones de requerimientos
 - 1.4.2 Análisis Orientado a Objetos
 - 1.4.3 Programación Orientada a Objetos y características
- 1.5 Elementos primordiales en el modelo de objetos
 - 1.5.1 Abstracción
 - 1.5.2 Encapsulamiento
 - 1.5.3 Modularidad
 - 1.5.4 Jerarquía y herencia
- 1.6 Historias de los paradigmas en el desarrollo de software

1.7 Beneficios del modelo de objetos y de la POO sobre otros paradigmas

UNIDAD II.- ARREGLOS UNIDIMENSIONALES Y MULTIDIMENSIONALES

TIEMPO APROXIMADO: 8 Horas

OBJETIVO DE LA UNIDAD: Aplicar la representación interna de los tipos de arreglos.

CONTENIDO

- 2.1 Arreglos Unidimensionales: listas (vectores)
 - 2.1.1 Conceptos básicos
 - 2.1.2 Operaciones
 - 2.1.3 Aplicaciones
- 2.2 Arreglo bidimensional
 - 2.2.1 Conceptos básicos
 - 2.2.2 Operaciones
 - 2.2.3 Aplicaciones
- 2.3 Arreglo Multidimensional
 - 2.3.1 Conceptos básicos
 - 2.3.2 Operaciones
 - 2.3.3 Aplicaciones

UNIDAD III.- MÉTODOS Y MENSAJES

TIEMPO APROXIMADO: 8 Horas

OBJETIVO DE LA UNIDAD: Aplicar los conceptos de las clases y los parámetros.

CONTENIDO

- 3.1 Atributos **const** y **static**.
- 3.2 Concepto de método
- 3.3 Declaración de métodos
- 3.4 Llamadas a métodos (mensajes)
- 3.5 Tipos de métodos
 - 3.5.1 Métodos **const**, **static**
 - 3.5.2 Métodos normales y volátiles
- 3.6 Referencia **tais**
- 3.7 Forma de pasar argumentos
- 3.8 Devolver un valor desde un método
- 3.9 Estructura del código

UNIDAD IV.- CONSTRUCTOR, DESTRUCTOR

TIEMPO APROXIMADO: 4 Horas

OBJETIVO DE LA UNIDAD: Aplicar el funcionamiento de los métodos de tipo constructor y destructor.

CONTENIDO

- 4.1 Conceptos de métodos constructor y destructor
- 4.2 Declaración de métodos constructor y destructor
- 4.3 Tipos de constructores y destructores
- 4.4 Aplicaciones de constructores y destructores

UNIDAD V.- SOBRECARGA

TIEMPO APROXIMADO: 4 Horas

OBJETIVO DE LA UNIDAD: Aplicar el concepto de sobrecarga de métodos y operadores.

CONTENIDO

- 5.1 Conversiones de tipos
- 5.2 Sobrecarga de métodos
- 5.3 Sobrecarga de operadores

UNIDAD VI.- HERENCIA

TIEMPO APROXIMADO: 8 Horas

OBJETIVO DE LA UNIDAD: Aplicar los conceptos de herencia

CONTENIDO

- 6.1 Introducción a la herencia
- 6.2 Herencia simple
- 6.3 Herencia múltiple
- 6.4 Clase base y clase derivada
 - 6.4.1 Definición
 - 6.4.2 Declaración
- 6.5 Parte protegida
 - 6.5.1 Propósito de la parte protegida
- 6.6 Redefinición de los miembros de las clases derivadas
- 6.7 Aplicaciones

UNIDAD VII.- POLIMORFISMO Y REUTILIZACIÓN

TIEMPO APROXIMADO: 8 Horas

OBJETIVO DE LA UNIDAD: Aplicar los conceptos de interfaces, clases abstractas y paquetes/librerías.

CONTENIDO

- 7.1 Concepto de polimorfismo
- 7.2 Clases abstractas
 - 7.2.1 Definición
 - 7.2.2 Redefinición
- 7.3 Definición de una interfaz
- 7.4 Implementación de la definición de una interfaz
- 7.5 Reutilización de la definición de una interfaz
- 7.6 Definición y creación de paquetes / librerías
- 7.7 Reutilización de las clases de un paquete / librería
- 7.8 Clases genéricas (plantillas)

UNIDAD VIII.- EXCEPCIONES

TIEMPO APROXIMADO: 8 Horas

OBJETIVOS DE LA UNIDAD:

- Identificar las condiciones de error que interrumpen el flujo normal de las sentencias en un programa.
- Utilizar el marco controlador del manejo de excepciones para desarrollar programas más seguros, estables y robustos.

CONTENIDO

8.1 Definición

8.1.1 Excepciones

8.2.1 Clases de excepciones, excepciones predefinidas por el lenguaje

8.3.1 Propagación

8.2 Gestión de excepciones

8.2.1 Manejo de excepciones

8.2.2 Lanzamiento de excepciones

8.3 Excepciones definidas por el usuario

8.3.1 Clase base de las excepciones

8.3.2 Creación de una clase derivada del tipo excepción

8.3.3 Manejo de una excepción definida por el usuario

UNIDAD IX.- FLUJOS Y ARCHIVOS

TIEMPO APROXIMADO: 8 Horas

OBJETIVO DE LA UNIDAD: Implementar aplicaciones Orientadas a Objetos que manipulen archivos de texto y binarios

CONTENIDO

9.1 Definición de archivos de texto y archivos binarios

9.2 Operaciones básicas en archivos texto y binario

9.2.1 Crear

9.2.2 Abrir

9.2.3 Cerrar

9.2.4 Lectura y escritura

9.2.5 Recorrer

9.3 Aplicaciones

EXPERIENCIAS DE APRENDIZAJE

- Uso de portal en Internet que cuente con un foro de discusión
- Utilizar software didáctico y de apoyo
- Ejercicios grupales fuera y dentro de clase.
- Formar equipos de trabajo para la exposición de investigaciones y tareas
- Fomentar el trabajo en equipo
- Creación de programas
- Solución de problemas
- Presentación de un proyecto final

CRITERIOS DE EVALUACIÓN

Exámenes por unidad

50%

Investigación y tareas	10%
Programas por computadora	20%
Proyecto Final	<u>20%</u>
TOTAL	100%

RECURSOS NECESARIOS

Recursos tecnológicos:

Visual C++, Java, Netbeans.

PRÁCTICAS SUGERIDAS

- Implementar clases y objetos cumpliendo las reglas de la programación orientada a objetos.
- Implementar clases con métodos y propiedades estáticas.
- Implementar clases con constructores y destructores del mismo objeto.
- Crear un programa que genere excepciones para identificar: nombre, causa, comportamiento y reporte de error.
- Programar una clase con varios métodos, donde el último método genere una excepción para estudiar y comprender la propagación de las mismas.
- Implementar aplicaciones orientadas a objetos que creen y manipulen archivos para guardar y recuperar información.
- Análisis, diseño e implementación de un proyecto de desarrollo basado en la Programación Orientada a Objetos; así como un Sistema de Información que acceda a una base de datos y herramientas tecnológicas que permitan el desarrollo de un proyecto mas integral (como lectores biométricos, código de barras, cámara web, entre otros).
- Investigación de distintos Lenguajes de Programación Orientados a Objetos.
- Discusión y comparación acerca de la Investigación de los distintos Lenguajes de Programación Orientados a Objetos.
- Desarrollo de prácticas aplicando los conocimientos de cada uno de los temas vistos.

BIBLIOGRAFÍA BÁSICA

BARNES, D., et al. (2007). Programación orientada a objetos con Java. Madrid: Pearson Educación.

BOOCH, G., et al. (2006). El lenguaje unificado de modelado. Madrid: Pearson Educación.

CEBALLOS, F. (2006). Java 2 curso de programación. México: Alfaomega.

BIBLIOGRAFÍA COMPLEMENTARIA

LEMAY, L., et al. (1999). Aprendiendo Java 2 en 21 días. México: Prentice Hall.

SCHILD, H. (2007). Fundamentos Java. México: McGraw_Hill.

LARMAN, Craig (1999). UML y patrones introducción al analisis y diseño orientado a objetos. México: Prentice Hall.

SMITH, M. (2000). Java: an object oriented language. UK: McGraw Hill.
BOOCH, G. (1996). Análisis y diseño orientado a objetos. México: Pearson Educación.
LEE, R. (2001). UML and C++: a practical guide to object oriented development. USA: Prentice Hall.
<http://www.javasoft.com>
<http://www.javaworld.com>