

Programa de Asignatura

Historia del programa

Lugar y fecha de elaboración	Participantes	Observaciones (Cambios y justificaciones)
Cancún, Q. Roo, 6/05/2010	M.C. Joel Antonio Trejo Sánchez	Se modificaron los objetivos y el contenido

Relación con otras asignaturas

Anteriores	Posteriores
Asignatura(s) Algoritmos y Estructura de datos Técnicas Algorítmicas	Asignatura(s) Arquitectura de Objetos Distribuidos
Tema(s) a) Estructuras de datos. b) Estructuras de control c) Recursividad	Tema(s) a) Clases b) Interfaces y Herencia

Nombre de la asignatura	Departamento o Licenciatura
Programación orientada a objetos	Ingeniería en Datos e Inteligencia Organizacional

Ciclo	Clave	Créditos	Área de formación curricular
2 - 2	IT0210	6	Profesional Asociado y Licenciatura Básica

Tipo de asignatura	Horas de estudio			
	HT	HP	TH	HI
Seminario	32	16	48	48

Objetivo(s) general(es) de la asignatura

Objetivo cognitivo

Describir las propiedades de la programación orientada a objetos para la distinción de sus principales características.

Objetivo procedimental

Construir aplicaciones utilizando el paradigma orientado a objetos para la solución a diversos problemas computacionales.

Objetivo actitudinal

Fomentar la disciplina en el análisis y solución de problemas para el desarrollo de habilidades.

Unidades y temas

Unidad I. CONCEPTOS BÁSICOS

Describir las principales características de la programación orientada a objetos para la implementación de soluciones utilizando este paradigma.

- 1) Tipos de datos abstractos
- 2) Clases
- 3) Objetos
- 4) Propiedades de la programación orientada a objetos
 - a) Abstracción
 - b) Encapsulamiento
 - c) Polimorfismo
 - d) Herencia

Unidad II. Abstracción y Encapsulamiento

Aplicar las propiedades de abstracción y encapsulamiento para la definición de clases e instanciación de objetos.

- 1) Modelando clases

2) Abstracción

3) Encapsulamiento

Unidad III. Herencia y Polimorfismo

Emplear las propiedades de herencia y polimorfismo para la definición de clases e instanciación de objetos.

1) Modelando clases y subclases

2) Herencia

a) Definición de superclases

b) Definición de subclases

3) Polimorfismo

Unidad IV. Interfaces y herencia múltiple

Usar interfaces y herencia múltiple para la implementación de programas utilizando el paradigma orientado a objetos.

1) Clases abstractas

2) Interfaces

3) Herencia múltiple

Unidad V. Programación Concurrente

Elegir el paradigma orientado a objetos para la creación de programas concurrentes.

1) Creación de hilos

2) Sincronización de hilos

3) Agrupación de hilos

Actividades que promueven el aprendizaje

Docente

Promover el trabajo individual en la definición de propuestas de solución a problemas determinados.
Coordinar la discusión de casos prácticos.
Aplicar prácticas para la definición de programas orientados a objetos.

Estudiante

Realizar tareas asignadas
Participar en el trabajo individual y en equipo
Resolver casos prácticos
Discutir temas en el aula
Participar en actividades extraescolares

Actividades de aprendizaje en Internet

El estudiante deberá acceder al portal (señalar las actividades que realizarán):

Se promoverá el uso de mecanismos asíncronos (correo electrónico, grupo de noticias, WWW y tecnologías de información) como medio de comunicación.

Criterios y/o evidencias de evaluación y acreditación

Criterios	Porcentajes
Examen	30
Tareas	30
Evidencias individuales	20
Evidencias grupales	20
Total	100

Fuentes de referencia básica

Bibliográficas

Cay S. Horstman. (2008) Core Java 2. Advanced Features. (8th Edición) EUA: Prentice Hall

Deitel M. (2002) Cómo programar en C++. (2da Edición). EUA: Pearson

Deitel M. (2004) Cómo programar en Java. (2da Edición). EUA: Pearson

Guardati Buemo Silva. (2007). Estructura de datos orientada a objetos: algoritmos en C++ (1era Edición). EUA; Pearson.

Luis Joyanes. (2002) Programación en Java 2 Algoritmos y Estructura de datos. (1era Edición). EUA; McGraw-Hill

Web gráficas

<http://www.sg.com.mx/>

Fuentes de referencia complementaria

Bibliográficas

Weiss. (2002) Data Structures and problema solving using Java. (2da Edición) EUA: Addison- Wesley

Web gráficas

No aplica

Perfil profesiográfico del docente

Académicos

Maestría en Ciencias de la computación, Ingeniero de Software

Docentes

Tener experiencia docente a nivel superior mínima de 3 años en ingeniería.

Profesionales

Tener experiencia en desarrollo de sistemas.