1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	**Fundamentos de programación**
Carrera:	**Ingeniería en Sistemas Computacionales**
Clave de la asignatura:	**SCM - 0414**
Horas teoría-horas práctica-créditos	**3-2-8**

2.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Toluca del 18 al 22 agosto 2003.	Representantes de la academia de sistemas y computación de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Ingeniería en Sistemas Computacionales.
Institutos Tecnológicos de: La Laguna, Mérida 23 agosto al 7 noviembre del 2003	Academia de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación.
Instituto Tecnológico de León 1 al 5 de marzo 2004.	Comité de consolidación de la carrera de Ingeniería en Sistemas Computacionales.	Definición de los programas de estudio de la carrera de Ingeniería en Sistemas Computacionales.

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

Anteriores	
Asignaturas	**Temas**
Ninguna .	

Posteriores	
Asignaturas	**Temas**
Programación orientada a objetos.	Arreglos unidimensionales y multidimensionales. Métodos y mensajes. Constructor, destructor. Sobrecarga. Herencia.

b). Aportación de la asignatura al perfil del egresado

Conoce y posee habilidad para desarrollar y programar la solución de problemas mediante el uso de los conocimientos de programación.

4.- OBJETIVO(S) GENERAL(ES) DEL CURSO

El estudiante analizará problemas y representará su solución mediante modelos orientados a objetos, diseñando los algoritmos para las funciones miembros y las aplicaciones que interactúan con el objeto, así como implementarlos en algún lenguaje de programación.

5.- TEMARIO

Unidad	Temas	Subtemas
1	Conceptos básicos del modelo orientado a objetos.	1.1 Reconocimiento de objetos y clases en el mundo real y la interacción entre ellos. 1.2 La abstracción y el encapsulamiento como un proceso natura. 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 Diseño Orientado a Objetos. 1.4.4 Programación Orientada a Objetos, conceptos 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. 1.5.5 Polimorfismo. 1.6 Historia de los paradigmas en el desarrollo del software. 1.7 Beneficios del modelo de objetos y de la POO sobre otros paradigmas.
2	Técnicas básicas de modelado de objetos.	2.1 Definición de clases, atributos, métodos y objetos. 2.2 El Modelo como resultado de la abstracción. 2.3 El UML como una herramienta de modelado de objetos. 2.4 Planteamiento del problema. 2.4.1 Analizar el enunciado del problema. 2.4.2 Identificar funciones del sistema.

5.- TEMARIO (Continuación)

		2.5 Análisis. 2.5.1 Descubrir objetos en el dominio del problema. 2.5.2 Identificar atributos de los objetos. 2.5.3 Identificar métodos en los objetos. 2.6 Introducción al diseño de la solución. 2.6.1 Representación gráfica de una clase. 2.6.2 Diagramas de interacción entre la aplicación y una clase. 2.6.3 Diagramas de estado de una clase.
3	Técnicas de diseño detallado.	3.1 Diseño algorítmico. 3.1.1 Elementos y reglas de la representación gráfica de los algoritmos. 3.1.2 Implementación de algoritmos secuenciales (utilizando notación algebraica). 3.2 Diseño algorítmico de las funciones.
4	Introducción a la programación.	4.1 Clasificación del software. 4.1.1 Software del sistema. 4.1.2 Software de aplicación. 4.2 Conceptos de la programación. 4.2.1 Definición de programa. 4.2.2 Definición de programación. 4.2.3 Definición de lenguaje de programación. 4.3 Datos. 4.3.1 Definición. 4.3.2 Tipos de datos. 4.3.3 Identificadores. 4.3.4 Almacenamiento, direccionamiento y representación en memoria. 4.3.5 Sistema de numeración binaria y hexadecimal.

5.- TEMARIO (Continuación)

		4.4 Operadores, operandos y expresiones. 4.5 Prioridad de operadores, evaluación de expresiones. 4.6 Estructura básica de un programa. 4.7 Proceso de creación de un ejecutable.
5	Implementación de la clase.	5.1 Modificadores de acceso (Public, Private). 5.2 Encapsulamiento de la clase. 5.3 El método como elemento de la comunicación. 5.3.1 Sintaxis. 5.3.2 Concepto de parámetro. 5.3.3 Parámetros de salida y de entrada. 5.3.4 El constructor. 5.3.5 El destructor.
6	Estructuras secuenciales y selectivas.	6.1 Modificadores de acceso (Public, Private). 6.2 Entrada y salida de datos. 6.3 Interacción de la aplicación y la clase. 6.4 Estructuras selectivas. 6.4.1 Selectiva simple (si). 6.4.2 Selectiva doble (si / de otro modo). 6.4.3 Selectiva anidada. 6.4.4 Selectiva múltiple. 6.4.5 Selectiva Intenta (try/catch).
7	Estructuras de repetición.	7.1 Repetir mientras Selectiva simple (si). 7.2 Repetir hasta. 7.3 Repetir desde.

6.- APRENDIZAJES REQUERIDOS

- Conocimientos básicos de operación de una computadora.

7.- SUGERENCIAS DIDÁCTICAS

- Uso de un portal de Internet para apoyo didáctico de la materia, el cual cuente por lo menos con un foro, preguntas frecuentes, material de apoyo y correo electrónico.
- Definir los lineamientos de documentación que deberán contener las tareas.
- Coordinar la realización de modelos orientados a objetos a partir de entidades del mundo real, utilizando ejemplos simples del entorno del estudiante.
- Mostrar al estudiante programas completos de menor a mayor grado de dificultad y con base en cada una de las instrucciones que los componen enseñar la sintaxis del lenguaje.
- Utilizar el aprendizaje basado en problemas, trabajando en grupos pequeños, para sintetizar y construir el conocimiento necesario para resolver problemas relacionados con situaciones reales.
- Solicitar al estudiante, la elaboración de los programas ejemplo en la computadora.
- Solicitar al estudiante propuestas de problemas a resolver y que sean significativas para él.
- Propiciar el uso de terminología técnica apropiada.
- Propiciar que el estudiante experimente con diferentes programas encontrados en revistas, Internet y libros de la especialidad, que lo lleven a descubrir nuevos conocimientos.
- Fomentar el trabajo en equipo.
- Elaborar en coordinación de los estudiantes una guía de ejercicios para actividades extra clase.

8.- SUGERENCIAS DE EVALUACIÓN

- Ponderar tareas.
- Participación y desempeño en el aula y el laboratorio.
- Dar seguimiento al desempeño en el desarrollo del programa (dominio de los conceptos, capacidad de la aplicación de los conocimientos en problemas reales, trasferencia del conocimiento).
- Desarrollo de un proyecto final que integre todas las unidades de aprendizaje.
- Participación en dinámicas grupales (mesas redondas , conferencias, lluvia de ideas, debate, entre otras).
- Actividades de auto evaluación.
- Exámenes departamentales.
- Cumplimiento de los objetivos y desempeño en las prácticas.

9.- UNIDADES DE APRENDIZAJE

UNIDAD 1.- Conceptos básicos del modelo orientado a objetos.

Objetivo Educacional	Actividades de Aprendizaje	Fuentes de Información
El estudiante conocerá los conceptos básicos del enfoque orientado a objetos y su aplicación a situaciones del mundo real.	• Analizar escenarios del mundo real que le sean familiares y detectar los objetos presentes y sus interacciones. • Analizar objetos comunes y detectar sus atributos y funciones ocultas y externas. • Buscar y seleccionar información sobre la complejidad del software. • Realizar un mapa conceptual referente al ciclo de vida del software. • Buscar y seleccionar información validando su fuente sobre el ciclo de vida del software. • Analizar los conceptos básicos del modelo orientado a objetos y realizar una síntesis. • Buscar información para que en base en ella pueda realizar un cuadro comparativo entre los distintos paradigmas señalando sus ventajas y desventajas.	1

UNIDAD 2.- Técnicas básicas del modelado de objetos.

Objetivo Educacional	Actividades de Aprendizaje	Fuentes de Información
Aprenderá los conceptos básicos de las técnicas del modelado de objetos.	• Buscar y seleccionar información del lenguaje UML referente al modelado de clases, diagramas de estado, de clase y de interacción. • Elaborar diagramas de clase, de interacción y de estado a distintos problemas sencillos.	1

UNIDAD 3.- Técnicas de diseño detallado.

Objetivo Educacional	Actividades de Aprendizaje	Fuentes de Información
Comprenderá el concepto de algoritmo. Conocerá la terminología y reglas relacionada con los algoritmos. Aplicara un lenguaje algorítmico gráfico.	• Elaborar una síntesis referente a la terminología y reglas relacionada con los algoritmos. • Diseñar una solución de problema utilizando los distintos tipos de algoritmos estudiados.	3

UNIDAD 4.- Introducción a la programación.

Objetivo Educacional	Actividades de Aprendizaje	Fuentes de Información
Comprenderá los conceptos básicos de la programación, y escribirá expresiones aritméticas y lógicas en un lenguaje de programación.	• Realizar un mapa conceptual sobre los tipos de software y los conceptos básicos de programación. • Realizar ejercicios de codificación de expresiones aritméticas y lógicas en un lenguaje de programación. • Buscar la información necesaria para Instalar y configurar el compilador del lenguaje de programación a utilizar • Compilar y ejecutar un programa modelo. • Realizar cambios en expresiones lógicas y algebraicas de un programa modelo y analizar los resultados obtenidos.	2,4,5,6

UNIDAD 5.- Implementación de la clase.

Objetivo Educacional	Actividades de Aprendizaje	Fuentes de Información
Conocerá la estructura de una clase y su implementación en un lenguaje de programación.	• Implementar clases a partir de modelos realizados en unidades anteriores. • Analizar de que forma afectan los modificadores de acceso a las clases, atributos y métodos y cuando es recomendable cada uno. • Realizar ejemplos de clase que requieran de funciones parametrizadas y analizar las distintas formas de paso de parámetros.	8,9,10,11,12,13

UNIDAD 6.- Estructuras secuenciales y selectivas.

Objetivo Educacional	Actividades de Aprendizaje	Fuentes de Información
Comprenderá el uso y funcionamiento de las estructuras secuenciales y selectivas y las implementará en el desarrollo de aplicaciones.	• Realizar una síntesis sobre el funcionamiento y aplicación de las estructuras secuenciales y selectivas. • Implementar clases que impliquen el diseño de algoritmos que requieran estructuras secuenciales y selectivas para probarlas en una aplicación.	2,4,5,6,8,10 11,12,13

UNIDAD 7.- Estructuras de repetición.

Objetivo Educacional	Actividades de Aprendizaje	Fuentes de Información
Comprenderá el uso y funcionamiento de las estructuras de repetición y las implementará en el desarrollo de aplicaciones.	• Realizar una síntesis sobre el funcionamiento y aplicación de las estructuras de repetición. • Implementar clases que impliquen el diseño de algoritmos que requieran estructuras de repetición para probarlas en una aplicación.	2,4,5,6,8,9,10 11,12,13

10. FUENTES DE INFORMACIÓN

1. Martín Flowler Kendall Scott. *UML Gota a Gota.* Addison Wesley.
2. Deitel y Deitel. *Como Programar en C++ cuarta Edición.* Prentice Hall.
3. Jean-Paul Tremblay, Richar B. Bunt. *Introducción a la Ciencia de Las Computadoras. Enfoque Algorítmico.* McGraw Hill.
4. Bjarne Storstrup. *Lenguaje de Programación C/C++.*
5. James P. Cohoon, Jack W. Davidson. *Programación y diseño en C++. Introducción a la programación y al diseño orientado a objetos.* McGraw Hill.
6. Kris Jamsa . *C/C++ Programación Exitosa.* Computec.
7. Francisco Chartre. *C++ Builder.* Anaya Multimedia.
8. Deitel Harvey, Deitel Paul. *Como Programar en JAVA 5a. Edición.* Pearson/Prentice may.
9. Luis Joyanes Aguilar. *Programación en JAVA 2 1ª Edición.* Mc Graw Hill.
10. Wang Paul S. *Java con Programación Orientada a Objetos y aplicaciones en la WWW. 1ª Edición.* Thomson Editores.
11. Hebert Schildt. *Fundamentos de Programación en JAVA 2 1a Edición.* Mc Graw Hill.
12. Agustín Froufe. *JAVA 2 Manual de Usuario y Tutorial 2ª. Edición.* Alfaomega RA-MA.
13. David Flanagan. *JAVA en Pocas Palabras 2ª. Edición.* Mc Graw Hill.

Referencias en Internet

[1] www.bibitec.org.mx
[2] www.sunmicrosystem.com

11. PRÁCTICAS

Unidad	Práctica	
1	1	Utilizando objetos del mundo real, detectar y documentar los elementos primordiales del modelo de objetos.
2	1	Utilizando UML obtener el diagrama de clases , el diagrama interacción y los diagramas de estado para un problema. Se pueden utilizar problemas presentados por el profesor o utilizar problemas del mundo real presentados por el estudiante.
3	1	Utilizando los diagramas de clases obtenidos en la práctica No. 2, elaborar algoritmos para representar el comportamiento de una clase.
4	1	Investigar el tipo de software que se utiliza en una organización y presentarlo mediante la clasificación del software vista en la unidad IV.
	2	Investigar el tipo de software que se utiliza en una organización y presentarlo mediante la clasificación del software vista en la unidad IV.
	3	Representar gráficamente el almacenamiento de diferentes datos simples proporcionados por el profesor.
	4	Elaborar ejercicios que impliquen el uso de operadores, operandos y expresiones.
5	1	Elaborar definición de clases utilizando un lenguaje de programación a partir de problemas proporcionados por el maestro.
6	1	Implementar aplicaciones que utilicen clases con comportamientos que impliquen el uso de estructuras secuenciales y expresiones aritméticas y lógicas.
	2	Implementar aplicaciones que utilicen clases con comportamientos que impliquen el uso de estructuras selectivas, haciendo uso de una herramienta de depuración de aplicaciones.
7	1	Implementar aplicaciones que utilicen clases con comportamientos que impliquen el uso de estructuras repetitivas, haciendo uso de una herramienta de depuración de aplicaciones