

1.- DATOS DE LA ASIGNATURA

| |
|---|
| Nombre de la asignatura: Programación I |
| Carrera: Licenciatura en Informática |
| Clave de la asignatura: IFM - 0430 |
| 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 Puebla del 8 al 12 septiembre 2003. | Representantes de la academia de sistemas y computación de los Institutos Tecnológicos. | Reunión nacional de evaluación curricular de la carrera de Licenciatura en Informática. |
| Instituto Tecnológico de: Cd. Juárez, Colima, Jiquilpan, Ocotlán, Pachuca, Tepic 13 septiembre al 28 de noviembre 2003. | Academia de de sistemas y computación. | Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación. |
| Instituto Tecnológico de Tepic 15 al 19 de marzo 2004. | Comité de consolidación de la carrera de Licenciatura en Informática. | Definición de los programas de estudio de la carrera de Licenciatura en Informática. |

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

| Anteriores | | Posteriores | |
|------------------------------|---|---|--|
| Asignaturas | Temas | Asignaturas | Temas |
| Fundamentos de programación. | Fundamentos del paradigma orientado a objetos. Construcción de comportamiento de un objeto Modelado de aplicaciones utilizando la POO | Estructura de datos Organización de datos. | Estructuras lineales. Estructuras no lineales. Listas enlazadas. Métodos de ordenamiento y búsqueda. Archivos. |

b). Aportación de la asignatura al perfil del egresado

- Proporciona conocimientos teórico - prácticos en la solución de problemas usando un lenguaje orientado a objetos.
- Representa un primer acercamiento al desarrollo de aplicaciones.

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

El estudiante comprenderá y aplicará los conceptos de la orientación a objetos en la construcción de programas utilizando un lenguaje de programación para la solución de problemas.

5.- TEMARIO

| Unidad | Temas | Subtemas |
|--------|-----------------------------------|--|
| 1 | Fundamentos del lenguaje. | <ul style="list-style-type: none">1.1 Introducción al lenguaje y a su entorno de desarrollo.1.2 Comentarios.1.3 Variables y constantes.1.4 Objetos que permiten E/S por consola.1.5 Operadores.1.6 Tipos de datos.<ul style="list-style-type: none">1.6.1 Fundamentales.1.6.2 Definidos por el usuario.1.7 Palabras reservadas.1.8 Expresiones.1.9 Estructuras de control.<ul style="list-style-type: none">1.9.1 Asignación.1.9.2 Selección.1.9.3 Iteración. |
| 2 | Subprogramas. | <ul style="list-style-type: none">2.1 Definición de un subprograma.<ul style="list-style-type: none">2.1.1 Estructura de un subprograma.2.1.2 Valor de retorno.2.2 Declaración de un subprograma.2.3 Bibliotecas o librería de subprogramas.2.4 Primer acercamiento a Clases y Objetos.2.5 Ámbito y tiempo de vida de variables.2.6 Argumentos y paso de parámetros.2.7 Sobrecarga de subprogramas.2.8 Recursividad. |
| 3 | Punteros, referencias y arreglos. | <ul style="list-style-type: none">3.1 Creación.3.2 Operaciones con punteros.3.3 Referencias3.4 Arreglos unidimensionales, bidimensionales y multidimensionales.3.5 Cadenas de caracteres.3.6 Asignación dinámica de memoria.3.7 Uso de clases predefinidas para arreglos. |

5.- TEMARIO (Continuación)

| | | |
|---|-------------------|---|
| 4 | Clases y Objetos. | <ul style="list-style-type: none">4.1 Definición de una clase.4.2 Declaración de clases.4.3 Miembros de una clase.4.4 Ámbito referente a una clase.4.5 Especificadores de acceso.4.6 Creación de objetos.4.7 Puntero this.4.8 Constructores y destructores. |
| 5 | Herencia. | <ul style="list-style-type: none">5.1 Importancia de la herencia en la POO5.2 Jerarquía de herencia.<ul style="list-style-type: none">5.2.1 Conceptos de Herencia simple y múltiple.5.2.2 Principios generales de diseño de jerarquías.5.2.3 Especificadores de acceso a jerarquía de clases.5.3 Definición de una clase base.5.4 Definición de una clase derivada<ul style="list-style-type: none">5.4.1 Constructores y destructores de clases derivadas.5.4.2 Conversión implícita de objetos de clase derivada a objeto de clase base.5.5 Herencia múltiple. |
| 6 | Polimorfismo. | <ul style="list-style-type: none">6.1 Concepto de polimorfismo.6.2 Clase base abstracta.6.3 Subprogramas virtuales.6.4 Destructores virtuales. |
| 7 | Archivos. | <ul style="list-style-type: none">7.1 Clases de E/S Clase base abstracta.7.2 Realizar entrada y salida de texto.7.3 Leer y escribir archivos.7.4 Realizar entrada y salida binaria. |

6.- APRENDIZAJES REQUERIDOS

- Conocimiento de los paradigmas de programación.
- Dominio de las metodologías para la solución de problemas mediante la construcción de algoritmos.
- Dominio de las técnicas de diseño de algoritmos.

7.- SUGERENCIAS DIDÁCTICAS

- Propiciar el trabajo en equipo.
- Desarrollar ejemplos de lo simple a lo complejo, buscando que el estudiante, asocie el tema con elementos significativos de su entorno.
- Realizar un taller donde el profesor conduzca paso a paso a sus estudiantes en el desarrollo del ejemplo, teniendo el cuidado que durante la solución se resalten los puntos de importancia que la teoría define, ya sea como concepto o el uso de la sintaxis en el momento de su aplicación.
- Elaborar un conjunto de problemas actuales asociados al entorno.
- Solicitar al estudiante el desarrollo de un proyecto de su interés, aplicando los conocimientos adquiridos durante el curso.
- Propiciar el desarrollo y la realización de prácticas.
- Involucrar al estudiante en las innovaciones que la evolución de la programación genera día con día, propiciando que éste se documente en distintas fuentes de información impresas y electrónicas.
- Seleccionar en acuerdo de academia, el lenguaje de programación a utilizar conforme a las necesidades del entorno, el contenido del programa y los recursos disponibles.

8.- SUGERENCIAS DE EVALUACIÓN

- Aplicar un examen inicial para conocer en nivel de los aprendizajes requeridos.
- Establecer de común acuerdo con los estudiantes, la ponderación de las diferentes actividades del curso.
- Participar en clase y en el taller.
- Exponer temas específicos en aula.
- Elaborar reportes de trabajos de investigación.
- Elaborar reportes de prácticas.
- Aplicar exámenes escritos y prácticos (en computadora) correspondientes a cada unidad.
- Dar seguimiento a los avances y documentación del proyecto propuesto.

9.- UNIDADES DE APRENDIZAJE

UNIDAD 1.- Fundamentos del lenguaje.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|--|----------------------------------|
| <p>El estudiante identificará la evolución, aplicaciones y sintaxis del lenguaje de programación, así como sus estructuras básicas.</p> <p>Identificará un problema que será utilizado para elaborar la propuesta de proyecto del curso.</p> | <p>1.1 Desarrollar una investigación en distintas fuentes y elaborar un informe donde identifique las principales etapas de la evolución que ha tenido el lenguaje a utilizar.</p> <p>1.2 Investigar y describir la sintaxis del lenguaje a utilizar.</p> <p>1.3 Configurar y manipular el entorno de desarrollo del lenguaje, resolviendo problemas que hagan énfasis a cada uno de los elementos de éste.</p> <p>1.4 Desarrollar un programa de aplicación en el cual globalice los puntos comprendidos en la unidad.</p> <p>1.5 Identificar el área de aplicación del lenguaje.</p> <p>1.6 Identificar un problema que pueda ser utilizado como proyecto de curso de acuerdo con el profesor.</p> | <p>1, 2, 3, 4, 5, 6, 7, 8, 9</p> |

UNIDAD 2.- Subprogramas.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|---|-------------------------------|
| <p>Conocerá el uso y aplicación de los subprogramas dentro de un lenguaje de programación.</p> <p>Desarrollará programas de aplicación relacionados con el uso de subprogramas.</p> | <p>2.1 Desarrollar programas que ejemplifiquen el uso de subprogramas definidos por el programador.</p> <p>2.2 Elaborar una guía rápida, en la que se describan los subprogramas de bibliotecas o librerías que proporciona el lenguaje de programación, propuestas por el profesor.</p> <p>2.3 Elaborar un programa que permita ejemplificar algunos de los subprogramas descritos en el punto anterior.</p> | <p>1, 2, 3, 4, 5, 6, 7, 8</p> |

| | | |
|--|--|--|
| | <p>2.4 Hacer analogías, por medio de hechos o acciones reales y cotidianas, del concepto de sobrecarga de subprogramas , presentarlo por escrito y discutirlo en el grupo.</p> <p>2.5 Desarrollar un programa en el cual se ejemplifique la sobrecarga de subprogramas.</p> <p>2.6 Presentar el avance del proyecto.</p> | |
|--|--|--|

UNIDAD 3.- Punteros, referencias y arreglos.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|---|-------------------------------|
| Desarrollará aplicaciones que permitan el empleo de la asignación estática y dinámica de memoria. | <p>3.1 Desarrollar ejemplos que permitan distinguir las diferencias entre: los operadores de creación, direccionamiento y de acceso al valor de una variable, de forma indirecta.</p> <p>3.2 Investigar y documentar el concepto de asignación dinámica de memoria.</p> <p>3.3 Desarrollar aplicaciones basadas en arreglos estáticos y dinámicos.</p> <p>3.4 Desarrollar aplicaciones basadas en cadenas de caracteres.</p> <p>3.5 Presentar el avance del proyecto.</p> | 1, 2, 3, 4, 5, 6, 7, 8 |

UNIDAD 4.- Clases y Objetos.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|--|-------------------------------|
| Aplicará los conceptos de la programación orientada a objetos usando el lenguaje. | <p>4.1 Identificar y hacer clasificaciones de objetos cotidianos y discutir en el aula los criterios seguidos para realizar tal clasificación.</p> <p>4.2 Determinar las características y operaciones de las clasificaciones analizadas e identificar el concepto de encapsulamiento.</p> <p>4.3 Desarrollar programas donde aplique la definición de clases y la construcción y manipulación de objetos.</p> | 1, 2, 3, 4, 5, 6, 7, 8 |

| | | |
|--|--------------------------------------|--|
| | 4.4 Presentar el avance del proyecto | |
|--|--------------------------------------|--|

UNIDAD 5.- Herencia.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|---|-------------------------------|
| Aplicará el mecanismo de la herencia simple y múltiple que provee el lenguaje. | 5.1 Describir, por medio de un informe, la importancia de la herencia en la programación orientada a objetos. 5.2 Solucionar problemas en los cuales aplique el análisis y el diseño de jerarquías de clases. 5.3 Discutir, en el grupo, los resultados del diseño de jerarquía y realizar los ajustes convenientes. 5.4 Desarrollar aplicaciones que sustenten la jerarquía planteada utilizando herencia simple. 5.5 Desarrollar aplicaciones que sustenten la jerarquía que utilizan herencia múltiple. 5.6 Presentar el avance del proyecto. | 1, 2, 3, 4, 5, 6, 7, 8 |

UNIDAD 6.- Polimorfismo.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|---|-------------------------------|
| Aplicará el concepto de polimorfismo por medio del lenguaje. | 6.1 Realizar un análisis comparativo de las definiciones de polimorfismo en distintas fuentes de información y elaborar una definición propia. 6.2 Desarrollar aplicaciones que utilicen, al menos, una clase base abstracta. 6.3 Presentar el avance del proyecto. | 1, 2, 3, 4, 5, 6, 7, 8 |

UNIDAD 7.- Archivos.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|--|------------------------|
| Manipulará archivos utilizando el lenguaje. | 7.1 Investigar todas las formas de manipular archivos por medio de objetos. 7.2 Desarrollar aplicaciones que permitan utilizar archivos. 7.3 Presentar proyecto final. | 1, 2, 3, 4, 5, 6, 7, 8 |

10. FUENTES DE INFORMACIÓN

1. H.M. Daitel/ P. J. Deitel.
Como programar en C/C++,
Ed. Prentice Hall Hispanoamericana.
2. Anthony Sintés.
Aprendiendo Programación Orientada A Objetos en 21 lecciones avanzadas con java.
Ed. Pearson Educación, México, 2002.
3. Bjarne Stroustrup.
El Lenguaje de programación C++.
Ed. Addison-Wesley/Díaz de Santos.
4. Bruce Eckel.
Aplique C++.
Ed. Mc. Graw Hill.
5. Herbert Schildt.
C# Manual de Referencia.
Ed. Mc. Graw Hill, 2003.
6. Templeman Jukian y Oslen Andy.
Microsoft Visual C++.NET aprenda ya.
Ed. McGraw-Hill. Profesional, Microsoft.
7. Schildt .
C++ Guía de auto enseñanza.
Ed. Mc-Graw Hill.

8. Herbert Schildt.
Fundamentos de Java 2.
Ed. Graw Hill, 2003.

Referencias en Internet

- [9] www.jugarijuegos.com./juegos/java
Desarrollo de estrategias de solución de problemas.

11. PRÁCTICAS

Unidad Práctica

- 1 Solución de problemas mediante un lenguaje de programación.
- 2 Elaborar pequeños proyectos para integrar los mecanismos orientados a objetos, de acuerdo al las sugerencias didácticas.
- 3 Elaborar la clase ventana.
- 4 Elaborar la clase menú.
- 5 Elaborar un editor de texto básico O.O.
- 6 Elaborar un sistema administrativo O.O
- 7 Incluir en las prácticas las propuestas en las actividades de aprendizaje.