

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: Programación II
Carrera: Ingeniería Electrónica
Clave de la asignatura: ECF-0432
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 Orizaba, del 25 al 29 de agosto del 2003.	Representante de las academias de ingeniería electrónica de los Institutos Tecnológicos.	Reunión Nacional de Evaluación Curricular de la Carrera de Ingeniería Electrónica.
Institutos tecnológicos de La Piedad, Nuevo Laredo y Tuxtla Gutiérrez, de septiembre a noviembre del 2003	Academias de Ingeniería Electrónica.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación
Instituto Tecnológico de Mexicali, del 23 al 27 de febrero 2004	Comité de consolidación de la carrera de Ingeniería Electrónica.	Definición de los programas de estudio de la carrera de Ingeniería Electrónica.

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
Programación I	<ul style="list-style-type: none"> - Diseño de algoritmos de programación. - Estructuras condicionales, incondicionales y repetitivas. - Funciones y procedimientos. 	Introducción a las Telecomunicaciones.	- Protocolos de comunicación.
		Modulo de Especialidad	

b). Aportación de la asignatura al perfil del egresado

- Programar en un lenguaje orientado a objetos.
- Fomentar el análisis lógico.
- Desarrollar una metodología de trabajo.
- Utilizar protocolos de comunicaciones electrónicas
- Adquirir habilidad para trabajo en equipo.

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

El estudiante diseñará y desarrollará aplicaciones con el paradigma de programación orientada a objetos.

5.- TEMARIO

Unidad	Temas	Subtemas
1	Fundamentos de la Programación orientada a objetos	1.1 Análisis del paradigma 1.2 Clases y objetos 1.3 Métodos y mensajes 1.4 Herencia 1.5 Polimorfismo 1.6 Encapsulamiento dinámico de la información
2	Programación en Lenguaje orientado a objetos	2.1 Desarrollo histórico del lenguaje orientado a objetos 2.2 Estructura general de un programa 2.3 Palabras reservadas 2.4 Representación de datos básicos 2.5 Definición de variables, Constantes y Funciones 2.6 Operaciones Aritméticas, Lógicas y de Manejo de caracteres 2.7 Declaración / Definición de clases 2.8 Definición/Activación de objetos de instancia 2.9 Diseño, Edición Compilación y Ejecución de programas.

5.- TEMARIO (Continuación)

Unidad	Temas	Subtemas
3	Programación visual y orientada a eventos	3.1 Paradigma de la programación visual 3.2 Paradigma de la Programación orientada a eventos 3.3 Lenguajes de programación visual y orientada a eventos 3.4 Diseño y construcción de programas visuales y orientados a eventos 3.5 Diseño de Interfases visuales 3.6 Aplicaciones
4	Enlace con simuladores	4.1 Enlaces de la programación orientada a objetos con simuladores
5	Programación de Puertos	5.1 Conceptos básicos de programación de puertos mediante objetos. 5.2 Configuración de los puertos locales en las computadoras. 5.3 Configuración de puertos remotos mediante protocolos de comunicación.
6	Activación Remota	7.1 Elaboración de un programa utilizando una interfase remota.

6.- APRENDIZAJES REQUERIDOS

- Diseño de algoritmos de programación.
- Estructuras condicionales, incondicionales y repetitivas.
- Funciones y procedimientos.

7.- SUGERENCIAS DIDÁCTICAS

- Propiciar la búsqueda y selección de información de los temas del curso.
- Exposición verbal auxiliado de materiales de apoyo.
- Fomentar la solución de problemas específicos aplicando el paradigma de programación orientado a objetos.
- Fomentar la implementación de programas, en forma autónoma, mediante prácticas.
- Promover en el alumno la realización de reportes y exposiciones auxiliado por software de presentaciones.
- Fomentar la implementación de programas orientados al manejo y uso del hardware mediante un lenguaje orientado a objetos para el análisis, solución y presentación de ejercicios, prácticas y proyectos.

8.- SUGERENCIAS DE EVALUACIÓN

- Considerar la participación en las actividades programadas de la materia:
 - Tareas y Ejercicios
 - Participación/Exposiciones en clase
 - Prácticas en clase
 - Prácticas extraclase
- Exámenes escritos y frente a la computadora
- Proyecto Final que involucre el manejo de hardware.

9.- UNIDADES DE APRENDIZAJE

Unidad 1: Fundamentos de la Programación Orientada a Objetos

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
El estudiante identificará y analizará los fundamentos de la programación orientada a objetos.	<ul style="list-style-type: none">• Investigar el desarrollo histórico del paradigma de la programación orientada a objetos. Comparar en grupo con la programación estructurada.• Identificar clases y objetos mediante ejemplos propuestos.• Identificar métodos y mensajes mediante ejemplos.• Analizar el concepto de herencia, polimorfismo y proponer ejemplos.• Analizar el concepto de encapsulamiento dinámico de la información y proponer ejemplos.	1 2

Unidad 2: Programación en Lenguaje Orientado a Objetos

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Diseñará programas básicos en un lenguaje orientado a objetos.	<ul style="list-style-type: none"> Investigar el desarrollo histórico del lenguaje orientado a objetos. Exponer sus resultados auxiliado de software de presentaciones. 	3
	<ul style="list-style-type: none"> Identificar la estructura general de un programa, las palabras reservadas y la representación básica de datos. 	4
	<ul style="list-style-type: none"> Identificar la definición de variables, constantes y funciones. 	5
	<ul style="list-style-type: none"> Identificar las operaciones aritméticas, lógicas y relacionales. 	6
	<ul style="list-style-type: none"> Identificar la definición / declaración de clases y la definición / activación de objetos de instancia. 	7
	<ul style="list-style-type: none"> Realizar el diseño, edición, compilación y ejecución de programas-ejercicios propuestos. 	8
		10
		11

Unidad 3: Programación visual y orientada a objetos

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información	
Aplicará la programación visual orientada a eventos en el diseño de interfases visuales.	<ul style="list-style-type: none"> Investigar el paradigma de la programación visual. Analizar en grupo las ventajas de este tipo de programación. 	3	
	<ul style="list-style-type: none"> Investigar el paradigma de la programación orientada a eventos. Analizar en grupo las ventajas de este tipo de programación. 	4	
	<ul style="list-style-type: none"> Analizar el lenguaje de programación visual y orientada a eventos que se utilizará. 	5	
	<ul style="list-style-type: none"> Diseñar y desarrollar programas visuales y orientados a objetos mediante ejercicios propuestos. 	6	
	<ul style="list-style-type: none"> Diseñar interfases visuales resolviendo ejercicios propuestos. 	7	
			8
			9
		10	
		11	

Unidad 4: Enlace con simuladores.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Realizará el enlace de un simulador con programas desarrollados en un lenguaje orientado a objetos.	<ul style="list-style-type: none">• Investigar las opciones del lenguaje de programación utilizado para enlazarse con simuladores. Realizar enlaces de programas con simuladores.	12

Unidad 5: Programación de puertos

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Diseñará programas para el manejo de puertos por medio de la programación orientada a objetos.	<ul style="list-style-type: none">• Identificar los conceptos básicos del manejo de puertos en la programación orientada a objetos.• Analizar la configuración de los puertos paralelo y serial en la computadora.• Diseñar programas para el manejo de puertos mediante prácticas propuestas.	3, 4 5, 6 7, 8 9, 10 11

Unidad 6: Activación Remota

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Diseñará e implementará un programa utilizando una interfase remota.	<ul style="list-style-type: none">• Diseñar e implementar un programa para activar un puerto mediante una interfase remota.	1, 2 3, 4 5, 6 7

10.- FUENTES DE INFORMACIÓN

- 1 Robert Lafore, *Object Oriented Programming*, Ed. Addison Wesley
- 2 Booch, Grady, *Object Oriented Analysis and Design*, Ed. The Benjamin/Cummings Publishing Company, Inc.
- 3 John Pew, *Instant Java*, Ed. Java Series, The Sunsoft Press
- 4 Emily Vander Veer, *Javascript for dummies*, Ed. IDG BOOKS
- 5 Emily Vander Veer, *Javascript for dummies*, Ed. IDG BOOKS
- 6 Harold Davis, *Delphi for windows*, Ed. Ventana
- 7 *Introducción a la programación en visual basic*, Ed. Mc Graw Hill
- 8 *Programación en Java*, Ed. Osborne
- 9 Petroutsos, Evagenlos, *La Biblia de Visual Basic .NET.*, Ed. Anaya Multimedia
- 10 Charre, Francisco, *Programación con Visual Basic .NET.*, Ed. Anaya Multimedia
- 11 Ferguson, Jeff, *La Biblia de C*, Ed. Anaya Multimedia
- 12 Cantú, Marco, *La Biblia de Delphi 7*, Ed. Anaya Multimedia
- 13 Dolores, M. Esther, *Solución de Problemas de Ingeniería con MATLAB*, Ed. Prentice Hall

10a.- Software Propuesto

- Visual Basic .NET
- Delphi
- Visual C++ .NET
- Builder C++
- Java++

11. PRÁCTICAS

Programación Orientada a Objetos

- Diseño de clases ("chips") para resolver problemas de electrónica.
- Ejemplo: realizar un programa que muestre una pantalla en la cual se encuentren botones con símbolos electrónicos, a los cuales cuando se presione el botón izquierdo del mouse aparezca una ventana que contenga la figura del dispositivo y su explicación.
- Hacer un simulador básico de circuitos eléctricos, donde cada elemento del circuito sea una clase diseñada.
- Hacer un editor gráfico básico aplicando herencia y polimorfismo.

Programación Orientada a Eventos

- Realizar un programa donde se utilice la clase que define una forma o ventana de Windows.
- Hacer una aplicación de múltiples ventanas.

Programación de puertos

- Diseñar un pequeño protocolo para emitir órdenes por puertos a una interfaz electrónica.
- Hacer un programa de transferencia de archivos de PC a PC.
- Hacer un programa para simular transferencia de bits y detección de errores en la transferencia.