



COMPUTACIÓN PARA INGENIEROS

Elaborado por	ING. MCS. YELITZA OVIEDO		AUTORIZADO POR VICE RECTORADO ACADÉMICO
Fecha de vigencia	ABRIL, 2005		
Revisado por	UNIDAD CURRICULAR..	DECANATO	

FUNDAMENTACION

El programa cubre los temas fundamentales para el inicio del estudio de programación en lenguaje de alto nivel (sugerencia Pascal).

El programa incluye tres unidades, a saber:

- I Unidad: Estructura Básica del Lenguaje.
- II Unidad: Modularidad.
- III Unidad: Manejo de Estructura de Datos.

Modalidad y Estrategias de Enseñanza. Conforme a la naturaleza del curso y a las modalidades y estilos de enseñanza sugeridos por la U.F.T., el docente empleará las siguientes modalidades y estrategias de enseñanza:

- Estrategia Cooperativa y colaborativa.
- Métodos asociados.
- Revisión de la literatura/controles de lectura.
- Material didáctico multimedia con explicaciones teórico-prácticas.
- Ejercicios de desempeño.
- Exposiciones modalidad video.
- Demostrativos.
- Preguntas y respuestas.

En la estrategia de estudio se recomienda una sólida preparación y revisión del material en el aula virtual. Además de las actividades de estudio individual, se sugiere formar foros interactivos para intercambiar notas y opiniones, para reforzar colectivamente las destrezas conceptuales y prácticas. De máxima importancia una constante dedicación por la resolución de ejercicios propuestos así como el uso del computador para correr los programas codificados.

OBJETIVO GENERAL DE LA ASIGNATURA

Desarrollar en el estudiante la habilidad para resolver en forma apropiada problemas científicos mediante el lenguaje de programación de alto nivel, considerando el manejo de estructuras básicas, modularidad y manejo de estructura de datos.

UNIDAD I		OBJETIVO TERMINAL	
ESTRUCTURA BÁSICA DEL LENGUAJE		DESARROLLAR PROGRAMAS DE COMPUTACIÓN DE MEDIANA COMPLEJIDAD CON EL PROPÓSITO DE EJEMPLIFICAR LAS APLICACIONES COMERCIALES Y CIENTÍFICAS, CONSIDERANDO LAS CARACTERÍSTICAS DE UN PROGRAMA, DATOS DE ENTRADA, PROCESOS Y SALIDAS.	
DURACION			
6 SEMANAS			
EVALUACION			
30 %			
OBJETIVOS ESPECIFICOS	CONTENIDO	ESTRATEGIAS DE INSTRUCCION	
1. Analizar las características de un programa. 2. Conocer las características del lenguaje C. 3. Identificar la estructura de un programa. 4. Clasificar e identificar los tipos de datos. 5. Utilizar las instrucciones de entrada y salida dado un problema. 6. Conocer los diferentes operadores aritmético,lógicos y relacionales y su importancia dentro de un programa. 7. Identificar las diferentes instrucciones de control: selectivas y repetitivas en un problema.	• Características de un programa: entradas procesos y salidas. Conceptos básicos del lenguaje de programación. • Historia y característidicas del Lenguaje C. • Estructura de un programa: # include, # define, función main, bloques. Análisis del enunciado. • Tipos de datos. Constantes y variables. • Expresiones aritméticas, lógicas y relacionales: operadores y sus prioridades. • Instrucciones de entrada y salida: diferencias entre ellas. Formatos. • Instrucciones condicionales: IF, SWITCH, BREAK, CONTINUE y repetitivas: WHILE, DO Y FOR.	PRESENCIAL	SEMIPRESENCIAL
		• Revisión de literatura/control es de lectura. • Explicaciones teórico-prácticas • Asignación de trabajos en el laboratorio. • Ejercicios de desempeño.	•Ejercicios de Desempeño. Ejemplos de aplicación. •Interacción en el aula virtual •Exposiciones modalidad video. •Materiales didácticos multimedia
ESTRATEGIAS DE EVALUACION:			
PRESENCIAL		SEMIPRESENCIAL	
• Evaluación parcial. • Ejercicios prácticos en laboratorio. • Taller en aula.		• Cuestionario en línea • Foros de discusión • Asignación de tareas • Videos, Blogs, Chat, Glosario, Wikis • Proyecto con Defensa • Evaluación Presencial	

UNIDAD II		OBJETIVO TERMINAL	
MODULARIDAD		DEFINIR UN PROBLEMA PARA EL DISEÑO, IMPLEMENTACION Y PROBACION DE PROGRAMAS DE MEDIANA COMPLEJIDAD, CONSIDERANDO EL CONCEPTO DE FUNCIONES Y EL USO DE BIBLIOTECAS DEL LENGUAJE.	
DURACION			
6 SEMANAS			
EVALUACION			
30%			
OBJETIVOS ESPECIFICOS	CONTENIDO	ESTRATEGIAS DE INSTRUCCION	
1. Analizar el concepto de modularidad y diseño descendente. 2. Definir: Función, ámbito de variables: locales y globales, importancia, ventajas y activación. 3. Identificar los tipos de parámetros. 4. Identificar las bibliotecas que posee el lenguaje. 5. Definir biblioteca, su estructura, activación. 6. Manejar las funciones que permiten validar. 7. Construir programas en forma modular utilizando funciones y bibliotecas.	- Definición de modularidad y diseño descendente. - Definición, declaración, creación y activación de funciones. - Alcance de variables. - Definición de biblioteca, su estructura, y activación.	PRESENCIAL	SEMIPRESENCIAL
		- Revisión de literatura/control es de lectura. - Explicaciones teórico-prácticas. - Asignación de trabajos en el laboratorio. - Ejercicios de desempeño	- Ejercicios de Desempeño. Ejemplos de aplicación. - Interacción en el aula virtual - Exposiciones modalidad video. - Materiales didácticos multimedia
ESTRATEGIAS DE EVALUACION:			
PRESENCIAL		SEMIPRESENCIAL	
- Evaluación parcial. - Ejercicios prácticos en laboratorio. - Taller en aula.		- Cuestionario en línea - Foros de discusión - Asignación de tareas - Videos, Blogs, Chat, Glosario, Wikis - Proyecto con Defensa - Evaluación Presencial	

UNIDAD III		OBJETIVO TERMINAL	
TIPOS DE DATOS ESTRUCTURADOS		IDENTIFICAR LAS ESTRUCTURAS DE DATOS ADECUADAS A UN PROBLEMA DADO PARA UN MANEJO EFICIENTE DE LAS MISMAS, IMPLEMENTANDO LAS FUNCIONES NECESARIAS Y CONSIDERANDO LAS FACILIDADES PREVISTAS POR EL LENGUAJE PARA CADA UNA DE ESAS ESTRUCTURAS DE DATOS.	
DURACION			
4 SEMANAS			
EVALUACION			
40 %			
OBJETIVOS ESPECIFICOS	CONTENIDO	ESTRATEGIAS DE INSTRUCCION	
1. Definir arreglo, tipos. 2. Utilizar las estructuras de datos adecuadas a un problema dado. 3. Conocer los algoritmos de búsquedas y ordenamiento de arreglo. 4. Definir arreglo de dos dimensiones. 5. Conocer las diferentes funciones que facilitan el manejo de caracteres. 6. Manejar las funciones que permiten validar caracteres. 7. Utilizar estructuras y uniones. 8. Utilizar archivos.	- Estructura de datos, tipos de arreglos. - Definición de arreglo de una dimensión, declaración, manipulación, índices. - Búsqueda y ordenamiento de arreglos, arreglo en paralelo. - Arreglo de dos dimensiones, búsqueda. - Validación y manejo de cadenas. - Declaración y uso de estructuras y uniones - Creación de archivos secuenciales.	PRESENCIAL	SEMIPRESENCIAL
		Revisión de literatura/control de lectura. • Explicaciones teórico-prácticas. • Asignación de trabajos en el laboratorio. • Ejercicios de desempeño.	•Ejercicios de Desempeño. Ejemplos de aplicación. •Interacción en el aula virtual •Exposiciones modalidad video. •Materiales didácticos multimedia
ESTRATEGIAS DE EVALUACION:			
PRESENCIAL		SEMIPRESENCIAL	
- Evaluación parcial. - Evaluación corta de arreglo. - Ejercicios prácticos en laboratorio. - Proyecto		- Cuestionario en línea - Foros de discusión - Asignación de tareas - Videos, Blogs, Chat, Glosario, Wikis - Proyecto con Defensa - Evaluación Presencial	

BIBLIOGRAFIA

Cairo Oswaldo. **Metodología de la Programación**. Tomo I y II. 1ra Edición. Editorial Alfaomega Grupo Editor S.A. de C.V. 1995.

Joyanes A. Luis. **Programación en C**. 1ra Edición. Editorial Mc Graw Hill. España. 1994.

Carretero J., García I., Fernández J., Calderon A. **El Lenguaje de Programación C. Diseño e Implementación de Programas**. Última Edición. Prentice Hall.

Langsam Y., Augenstein M., Tenenbaum. **Estructuras de Datos con C y C++**. 2da edición. Prentice Hall. 1997.

Sedgewick Robert. **Algoritmo en C++**. Última Edición. Prentice Hall.

Antonakos J. Mansfield Jr., Hull J. **Programación Estructura en C**. Prentice Hall. 1997.

Deitel H., Deitel P. **Como Programar en C/C++**. 2da edición. Prentice Hall. 1995.

Levine Guillermo. **Computación y Programación Moderna**. Perspectiva Integral de la Informática. Prentice Hall. México. 2001.

Kernighan Brian, Ritchie Dennis. **El Lenguaje de Programación C**. 2da edición. Prentice Hall.

Kernighan Brian. **La Práctica de la Programación**. 6ta edición 2000. Prentice Hall.