



**UNIVERSIDAD FERMÍN TORO
VICE RECTORADO ACADÉMICO
FACULTAD DE INGENIERÍA**

PROGRAMA INSTRUCCIONAL

DATOS BÁSICOS DE LA ASIGNATURA							
Nombre de la asignatura:		Código	Semestre		U.C.	Pre- Requisito	
COMPUTACIÓN PARA INGENIEROS		COI-331	III		3	INI-221	
Tipo de Asignatura:	OBLIGATORIA	DENSIDAD HORARIA					
		Presencial			Semi presencial		THS/SEM
		H.T.	H.P/H.L	H.A	H.V	H.P	5/80
Escuela:	COMPUTACIÓN ELÉCTRICA MMTO.MECANICO TELECOM.	2	0	3	48	32	
DATOS BÁSICOS DEPARTAMENTO/S RESPONSABLE/S					ELABORADO POR:		
DEPARTAMENTO/S:	PROGRAMACIÓN				Ing. Esteban Torrealba		
DATOS BÁSICOS DE LA AUTORIZACIÓN							
FECHA DE VIGENCIA		REVISADO POR DECANATO		AUTORIZADO POR VICE RECTORADO ACADÉMICO			
SEPTIEMBRE, 2015		(FIRMA Y SELLO)		(FIRMA Y SELLO)			

FUNDAMENTACIÓN

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: | Introducción a la programación orientada a objetos |

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.
Métodos asociados.
Revisión de la literatura/controles de lectura.
Explicaciones teórico-prácticas.
Ejercicios de desempeño.
Expositivos.
Demostrativos.
Preguntas y respuestas.

En la estrategia de estudio se recomienda una sólida preparación y revisión previa a cada sesión de aquellos temas que serán tratados. Además de las tareas de estudio individual, se sugiere formar grupos de estudio estables 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 en clase 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
DURACIÓN		
6 SEMANAS		
EVALUACIÓN		
30 %		
OBJETIVOS ESPECÍFICOS	CONTENIDO	ESTRATEGIAS DE INSTRUCCIÓN
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 aritméticos, lógicos y relacionales y su importancia dentro de un programa. 7. Identificar las diferentes instrucciones de control: selectivas y repetitivas en un problema. 8. Definir arreglo, tipos. 9. Utilizar las estructuras de datos adecuadas a un problema dado. 10. Conocer los algoritmos de búsquedas y ordenamiento de arreglo. 11. Definir arreglo de dos dimensiones.	• Características de un programa: entradas procesos y salidas. Conceptos básicos del lenguaje de programación. • Historia y características del Lenguaje C. • Estructura de un programa: # incluye, # 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. • 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.	Presencial • Revisión de literatura. • Explicación teórica – práctico. Dinámica de grupo. Semi-Presencial • Ejercicios de Desempeño. Ejemplos de aplicación. • Interacción en el aula virtual • Exposiciones modalidad video. • Materiales didácticos multimedia
ESTRATEGIAS DE EVALUACIÓN:		
Presencial	Semi-Presencial	
• Taller. • Evaluación corta.	• Cuestionario en línea • Foros de discusión • Asignación de tareas • Videos, Blogs, Chat, Glosario, Wikis, juegos • Proyecto con Defensa • Evaluación Presencial	

UNIDAD II		OBJETIVO TERMINAL	
MODULARIDAD		DEFINIR UN PROBLEMA PARA EL DISEÑO, IMPLEMENTACIÓN Y PROBABIÓN DE PROGRAMAS DE MEDIANA COMPLEJIDAD, CONSIDERANDO EL CONCEPTO DE FUNCIONES Y EL USO DE BIBLIOTECAS DEL LENGUAJE.	
DURACIÓN			
5 SEMANAS			
EVALUACIÓN			
30 %			
OBJETIVOS ESPECÍFICOS	CONTENIDO	ESTRATEGIAS DE INSTRUCCIÓN	
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. 8. Conocer las diferentes funciones que Facilitan el manejo de caracteres. 9 Manejar las funciones que permiten validar caracteres. 10. Utilizar estructuras y uniones. 11. Utilizar archivos	Definición de modularidad y Diseño descendente. • Definición, declaración, creación y activación de funciones. • Alcance de variables. • Definición de biblioteca, su estructura, y activación Validación y manejo de cadenas. • Declaración y uso de estructuras y uniones • Creación de archivos secuenciales.	Presencial • Revisión de literatura. • Explicación teórica. • Taller teórico – práctico. • Dinámica de grupo. Semi-Presencial • Ejercicios de Desempeño. Ejemplos de aplicación. • Interacción en el aula virtual • Exposiciones modalidad video. • Materiales didácticos multimedia	
ESTRATEGIAS DE EVALUACIÓN:			
Presencial		Semi-Presencial	
• Taller. • Evaluación corta.		• Cuestionario en línea • Foros de discusión • Asignación de tareas • Videos, Blogs, Chat, Glosario, Wikis, juegos • Proyecto con Defensa • Evaluación Presencial	

UNIDAD III		OBJETIVO TERMINAL
INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS		APLICAR LOS CONCEPTOS BÁSICOS DE LA POO, LA TÉCNICA DE MODELADO ORIENTADO A OBJETOS Y LA REPRESENTACIÓN DE LAS CLASES EN UN LENGUAJE DE PROGRAMACIÓN
DURACIÓN		
5 SEMANAS		
EVALUACIÓN		
40 %		
OBJETIVOS ESPECÍFICOS	CONTENIDO	ESTRATEGIAS DE INSTRUCCIÓN
1. Aplicar los conceptos básicos de la Programación Orientada a Objetos (POO). 2. Identificar los pilares fundamentales de la programación orientada a objetos 3. Identificar los componentes básicos de un diagrama de clases. 4. Aplicar una técnica de modelado del mundo real mediante objetos, para la resolución de problemas, siguiendo una notación UML. 5. Emplear para un lenguaje de programación la representación de las clases diagramadas en UML. 6. Implementar Programación Orientada a Objetos usando la metodología de diseño.	1. Conceptos Básicos de POO Programación orientada a objetos Objetos, Clases, Instancias, Atributos, Métodos, Mensajes 2. Pilares fundamentales de la programación orientada a objetos - Abstracción de datos - Encapsulamiento de Información - Ocultamiento de la información - Herencia - Polimorfismo Lenguaje de Modelado Unificado - Definición de lenguaje unificado de modelado (UML) - Diagramas de clases en UML - Relaciones entre clases 4. Representación de clases en el lenguaje de programación - Definición de lenguajes de programación. - Definición de traductores, diferencias entre ensambladores, interpretadores y compiladores. - Definición y propósito de los lenguajes naturales y artificiales. Diferencias entre lenguajes naturales y artificiales.	Presencial - Revisión de literatura. - Explicación teórica – práctico. - Dinámica de grupo.
		Semi-Presencial - Ejercicios de Desempeño. Ejemplos de aplicación. - Interacción en el aula virtual - Exposiciones modalidad video. - Materiales didácticos multimedia
ESTRATEGIAS DE EVALUACIÓN:		
Presencial	Semi-Presencial	
- Taller. - Evaluación corta.	- Cuestionario en línea - Foros de discusión - Asignación de tareas - Videos, Blogs, Chat, Glosario, Wikis, juegos - Proyecto con Defensa - Evaluación Presencial	

BIBLIOGRAFÍA

- BOOCH, G. **Análisis y Diseño Orientado a Objetos con Aplicaciones**. Addison –Wesley. 1996.
- BUJANDA L. **Introducción a la Programación Orientada a Objetos utilizando C++**. Trabajo de Ascenso. 2004.
- COAD, P y YOURDON, E. 1991. **Object – Oriented Analysis**. Second edition, Prentice-Hall; Object - Oriented Design. Prentice-Hall
- DEVIS, R. **Programación Orientada a Objetos en C++**. Paraninfo S.A. 1993.
- DIAZ, R **Lineamiento para La Programación Orientada a Objetos, utilizando la Arquitectura Modelo Vista Controlador y el Lenguaje C++**. Trabajo de ascenso. UCLA.
- JOYANES, L y ZAHONERO, I. **Programación en C, C++, JAVA y UML**. Mc Graw Hill, Interamericana Editores, S.A. DE C.V. 2010.
- JOYANES, L. **Programación en C++. Algoritmos, estructuras de datos y objetos**. 2a Edición. Mc Graw Hill, Interamericana de España, S.A.U. 2006
- JOYANES, L. **Programación Orientada a Objetos**. 2a Edición. Mc Graw Hill, Interamericana de España, S.A.U. 1998.
- FUENTEZ, B. **Curso de Programación Orientada a Objetos**. 2003.
<http://www.abcdatos.com/tutoriales/tutorial/15937.html>.
- GARCÍA, A. **Conceptos Básicos de la Programación Orientada a Objetos**. 2002
<http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/clases1/clases.htm>.
- IZQUIERDO, L. **Introducción a la Programación Orientada a Objetos**. 2007
<http://luis.izqui.org/resources/ProgOrientadaObjetos.pdf>.