

DATOS DE IDENTIFICACIÓN

MATERIA:	PROGRAMACIÓN II				
CENTRO ACADÉMICO:	CENTRO DE CIENCIAS BÁSICAS				
DEPARTAMENTO ACADÉMICO:	SISTEMAS ELECTRÓNICOS				
PROGRAMA EDUCATIVO:	INGENIERIA EN SISTEMAS COMPUTACIONALES				
AÑO DEL PLAN DE ESTUDIOS:	2016	SEMESTRE:	3	CLAVE DE LA MATERIA:	24936
ÁREA ACADÉMICA:	REDES Y COMUNICACIONES		PERIODO EN QUE SE IMPARTE:	AGOSTO-DICIEMBRE	
HORAS SEMANA T/P:	3/2		CRÉDITOS:	8	
MODALIDAD EDUCATIVA EN LA QUE SE IMPARTE:	PRESENCIAL		NATURALEZA DE LA MATERIA:	OBLIGATORIA	
ELABORADO POR:	GSP, ELMA, ESP, BGER, ABA, JJPG				
REVISADO Y APROBADO POR LA ACADEMIA DE:	REDES Y PROGRAMACION DE SISTEMAS		FECHA DE ACTUALIZACIÓN:	JULIO 2022	

DESCRIPCIÓN GENERAL

Curso teórico-práctico, en el cual se abordan de manera progresiva las técnicas de programación orientada a objetos, el alumno entenderá inicialmente el proceso de abstracción y como éste se ve involucrado en el diseño de clases, sobrecarga de operadores, herencia y polimorfismo. En el proceso de cambio de paradigma el alumno adquiere la habilidad para la construcción de tipos de datos abstractos empleando dichas técnicas de programación avanzada a problemas complejos de diseño. Esta materia es posterior a Lógica de Programación y Programación I, se cursa simultáneamente con Estructuras de Datos. Los conocimientos de esta materia resultan fundamentales como antecedentes para Programación III.

OBJETIVO (S) GENERAL (ES)

Al finalizar el curso, el alumno evaluará las mejores opciones de programación avanzada disponibles para desarrollar soluciones computacionales a problemas complejos; mediante los datos abstractos, la sobrecarga de funciones, la herencia y polimorfismo, ya que resulten la opción más apropiada para solución de problemas complejos, de manera proactiva e innovadora, así como con calidad.

CONTENIDOS DE APRENDIZAJE

UNIDAD TEMÁTICA I: CONCEPTOS BÁSICOS DEL LENGUAJE C++ (20 horas aprox.)		
OBJETIVOS PARTICULARES	CONTENIDOS	FUENTES DE CONSULTA
<p>Comprender y aplicar los conceptos básicos para el manejo de entrada y salida de datos.</p> <p>Conocer los operadores de manejo de memoria dinámica para manipular arreglos y estructuras dinámicas en C++.</p> <p>Conocer y aplicar las distintas formas de apertura y manipulación de archivos, empleando archivos secuenciales y binarios en C++.</p>	<ol style="list-style-type: none"> Introducción a C++ <ol style="list-style-type: none"> El entorno del lenguaje Conceptos generales Estructura básica de un programa Flujos de entrada y salida de datos (cin/cout) Manejo de memoria dinámica <ol style="list-style-type: none"> Operadores new y delete Arreglos dinámicos Estructuras dinámicas Punteros a funciones Manejo de argumentos en la función main Archivos en C++ <ol style="list-style-type: none"> Flujo de entrada/salida (librería <i>fstream</i>) Manejo de archivos de texto y binarios 	1, 2, 3, 4

UNIDAD TEMÁTICA II: PROGRAMACION ORIENTADA A OBJETOS (15 horas aprox.)		
OBJETIVOS PARTICULARES	CONTENIDOS	FUENTES DE CONSULTA
<p>Conocer los fundamentos de la programación orientada a objetos que le permita manejar la abstracción de datos para construir clases.</p> <p>Entender el acceso a variables y funciones públicas y privadas dentro de una clase, así como el uso y la importancia del constructor, destructor y su sobrecarga.</p> <p>Aprender el concepto de apuntador a un objeto (this), y su uso con las variables y funciones miembros de una clase.</p> <p>Aprender la importancia y el uso adecuado de las funciones especiales.</p>	<ol style="list-style-type: none"> Conceptos de programación orientada a objetos <ol style="list-style-type: none"> Definición de clase, atributo, método y objeto Abstracción de datos <ol style="list-style-type: none"> Tipos de datos definidos por el usuario Tipos de datos abstractos Construcción de clases y objetos <ol style="list-style-type: none"> Componentes de una clase Datos y funciones miembro Implementación de funciones miembro Constructor y destructor Tipos de Constructores (por defecto, copia, sin/con argumentos) Ámbito y acceso a los miembros de una clase <ol style="list-style-type: none"> <i>public, private, protected, default</i> Puntero <i>this</i> Clases y estructuras Clases internas Creación y destrucción de objetos dinámicos Funciones miembro especiales <ol style="list-style-type: none"> Funciones <i>inline</i> Funciones y datos <i>static</i> y <i>const</i> Funciones y clases <i>friend</i> 	1, 2, 3, 4

UNIDAD TEMÁTICA III: SOBRECARGA DE FUNCIONES Y OPERADORES (15 horas aprox.)		
OBJETIVOS PARTICULARES	CONTENIDOS	FUENTES DE CONSULTA
<p>Aplicar la sobrecarga de funciones y operadores para optimizar la programación.</p> <p>Entender las diferencias entre los diversos operadores a sobrecargar.</p> <p>Desarrollar una aplicación que involucre los conceptos de sobrecarga de funciones y operadores.</p>	<ol style="list-style-type: none"> Sobrecarga de funciones <ol style="list-style-type: none"> Declaración Funciones no miembros sobrecargadas Funciones miembro sobrecargadas Sobrecarga de funciones amigas Sobrecarga de operadores <ol style="list-style-type: none"> Prioridad de operadores Restricciones en los operadores Declaración Operadores binarios y unarios Funciones amigas en operadores Operadores de flujo (<i>cin/cout</i>) Ejercicios con sobrecarga de operadores 	1, 2, 3, 4

UNIDAD TEMÁTICA IV: HERENCIA Y POLIMORFISMO (20 horas aprox.)		
OBJETIVOS PARTICULARES	CONTENIDOS	FUENTES DE CONSULTA
<p>Comprender y aplicar el concepto de herencia y polimorfismo para optimizar el diseño de herramientas de software.</p>	<ol style="list-style-type: none"> Herencia <ol style="list-style-type: none"> Clase base y clase derivada Miembros protected Uso de funciones miembro Herencia public, private y protected Uso de constructores y destructores en clases derivadas Sobreposición (override) de miembros de clase base en clase derivada Herencia múltiple Composición en comparación con la herencia Polimorfismo <ol style="list-style-type: none"> Funciones virtuales Funciones virtuales puras y clases base abstractas Destructores virtuales Clases abstractas Ligado dinámico 	1, 2, 3, 4

METODOLOGÍA DE ENSEÑANZA - APRENDIZAJE

- Exposiciones verbales por parte del profesor, de acuerdo a los temas establecidos en el programa de estudios, apoyándose en la bibliografía del mismo.
- Realización de un número suficiente de ejercicios frente a grupo.
- Se usará ANSI C++ como herramienta de apoyo en la solución de problemas.
- Realización por parte de alumnos de ejercicios, exposiciones y/o presentaciones, trabajos extra clase, entre otros, para verificar el dominio de los temas vistos en clase.
- Realización de ejercicios en el aula por parte de los alumnos, donde se apliquen los conocimientos adquiridos en la clase.

6. Juicio crítico del profesor que le permita en caso de falta de tiempo seleccionar los contenidos y objetivos básicos de cada unidad; previo visto bueno del coordinador de academia correspondiente.
7. Juicio crítico del profesor que le permita en caso contar con tiempo extra para seleccionar temas de interés para el grupo; previo visto bueno del coordinador de academia correspondiente.

RECURSOS DIDÁCTICOS

Tradicionales: pizarrón, material de apoyo, notas de la materia

Nuevas Tecnologías: Proyector, Computadora, Plataforma de ámbito académico

EVALUACIÓN DE LOS APRENDIZAJES

1er. Parcial	25%
2º. Parcial	25%
Examen Final	30%
Proyecto Final	20%
TOTAL	100%

Notas:

1. El lenguaje a manejar es cualquier variante de ANSI C/C++
2. La calificación de cada parcial se integra de actividades sumativas.
3. La calificación final de cada parcial será reportada en el sistema de acuerdo como marca el reglamento (7 días naturales a partir del fin del período de exámenes).

FUENTES DE CONSULTA

BÁSICAS:

1. Deitel, Harvey & Deitel, Paul. Cómo programar en C/C++. Prentice Hall, 2ª. Edición, 1994.
2. Deitel, Paul. C++ how to program: late objects version. Pearson. 7ª Edición, 2011
3. Garrido Carrillo, Antonio. "Fundamentos de Programación en C++". Delta Publicaciones, 1ª. Edición, 2013
4. Joyanes Aguilar, Luis. Fundamentos de Programación: Algoritmos, Estructura de Datos y Objetos. McGraw Hill. 4ª Edición, 2008

COMPLEMENTARIAS:

5. De Giusti, Armando E. "Algoritmos, datos y programas". Prentice-Hall, 2001
6. Kernighan, Brian W.; Ritchie, Dennis M. El Lenguaje De Programación "C". Segunda Edición. Editorial Prentice-Hall Hispanoamericana. Año 1995
7. Peñaloza Romero, Ernesto. Fundamentos De Programación C/C++. Delta publicaciones. 4ª Edición, 2005.
8. Prata, Stephen. C++ primer plus. Indianapolis. 4ª Edición, 2001
9. Ramírez, Felipe. Introducción a la programación: algoritmos y su implementación en VB.NET, C#, Java y C++. Alfaomega. 2ª Edición, 2007
10. Sznajdleder Augusto, Pablo. Algoritmos a fondo: con implementaciones en C y Java. Alfaomega. 1ª Edición, 2012

