



Universidad de Puerto Rico
Recinto de Mayagüez
Facultad de Artes y Ciencias
DEPARTAMENTO DE CIENCIAS MATEMATICAS
Programa de Ciencias de Computadoras

Curso: Introducción a la Programación de Computadoras II

Codificación: COMP 3110

Número de horas/crédito: 3

Prerrequisitos, correquisitos y otros requerimientos: Introducción a la Programación de Computadoras I (COMP3010) o su equivalente.

Información del profesor:

Nombre	
Horas de Oficina	
Oficina	
Ext.	
Dirección Electrónica	

Descripción del Curso:

Diseño, verificación y prueba de programas extensos. Técnicas de programación incluyendo recursión, programación orientada a objetos, uso de apuntadores y arreglos dinámicos. Pasos en el Ciclo de Desarrollo de Software, técnicas básicas de búsqueda y ordenamiento. Introducción de estructuras estáticas y dinámicas de Datos especialmente pilas, colas y listas.

Objetivos del Curso: Al finalizar el curso el estudiante alcanzará las siguientes metas:

- Continuar con la programación orientada a Objeto usando un lenguaje de alto nivel.
- Manejar los conceptos de Herencia, Polimorfismo, Encapsulación relacionados con clases
Conocer y analizar la programación orientado a Objeto
- Comparar las estructuras estáticas y dinámicas de Datos
- Conocer a fondo las construcciones del lenguaje de alto nivel seleccionado en el diseño de objetos.
- Aprender a través del lenguaje de alto nivel a dividir sus programas en subprogramas como entidades separadamente ejecutables.
- Aprender los beneficios o consecuencias generados por la recursión.
- Manejar estructuras dinámicas de Datos como herramientas importantes en el desarrollo de programas. Entre las estructuras estudiadas se incluyen la lista simple o doblemente enlazadas.
- Definir funciones genéricas ('templates') y su beneficio en la programación.

Bosquejo de contenido y distribución del tiempo :

LECCION	SECCION	TEMA
1		Introducción al curso
2-3	8.1, 8.2, 8.3	Repaso de arreglos de una dimensión. Definición de datos estructurados ("struct")
4	7.1 a 7.4	Repaso de funciones definidas por el programador. Pase de parámetros por valor y referencia
5,6,7	10.1, 10.2, 10.5	Repaso de manejo de archivos. Repaso de archivos como parámetros de funciones. <i>Definición de asignación #1</i>
8, 9	1.3, 2.3, 2.4, 2.5	Definición de clases. Declaración de atributos y métodos de una clase (member functions, private and public member functions).
10,11	3.1, 3.2, 3.3	Implantación de clases. <i>Definición de Asignación #2.</i>
12,13,14	7.5	Definición de POINTER y Arreglos Dinámicos. <i>Definición de Asignación #3.</i>
15,16,17	7.6	Operaciones con Arreglos Dinámicos. Comparaciones entre asignación estática y dinámica de memoria.
18,19,20	15.1, 15.2	'Templates'. Usos de arreglos con templates.
21,22,23	13.3	'Templates'. <i>Definición de Asignación #4.</i>
24,25,26	13.4	'Stacks'. Operaciones con Stack. Implementación usando arreglos. <i>Definición de Asignación #5.</i>
27		<i>EXAMEN #1</i>
28,29,30	13.4	'Queues'. Definición y operaciones y usos de Queues. <i>Definición de Asignación #6.</i>
31,32,33	13.2	'Linked Lists'. <i>Definición de Asignación #7.</i>
34,35,36	13.2	"Double Linked Lists, Tree". 'Simulación de estructuras estáticas usando listas enlazadas. <i>Definición de Asignación #8.</i>
37		<i>EXAMEN #2</i>

Estrategias instruccionales:

- Conferencias y uso de ayudas visuales como por ejemplo transparencias, presentaciones en 'power point' a través de WebCT.
- Laboratorio de computadoras donde se desarrollan los programas durante clases.
- Uso de dos Laboratorios de soporte para desarrollo de asignaciones fuera de clases que requieran la ejecución de programas.
- Uso del www para acceder la página del curso donde se encuentran ejemplos, ejercicios adicionales (laboratorios) información importante.
- Actividades centradas en el estudiante, como por ejemplo, modificar y/o escribir programas durante el periodo de laboratorio.

Recursos de aprendizaje o instalaciones mínimos disponibles o requeridos :

- Lenguaje de alto nivel instalado en Laboratorio de Computadoras bajo el sistema operativo Solaris.
- Otros Laboratorios bajo Solaris o Linux de apoyo en la solución de programas complementarios que facilitan el aprendizaje del estudiantado.

- Internet en todos los Laboratorio usados por el estudiantado.

Estrategias de evaluación: Exámenes escritos, quizzes escritos, quizzes prácticos, proyectos

Número de exámenes	Mínimo 3 exámenes incluyendo el final (60%)
Programas escritos en C++	8 programas (40%)

Sistema de calificación:

A	B	C	D	F
90% - 100%	80% - 89%	65% - 79%	60% - 64%	0% - 59%

Período de los exámenes finales: del 14 al 22 de mayo de 2009

Bibliografía

- **Problem Solving with C++, 7th edition,**
By Walter Savitch, Addison Wesley, 2009
- **Object-Oriented Program Development Using C++.**
By Gary J. Bronson.
Thomson, 2006.
- **Problem Solving with C++, 6th edition,**
By Walter Savitch, Addison Wesley, 2007

CH/gjd

14 de enero de 2009