

CI-1201 Programación 2

Plan: Bachillerato	Créditos: 4	g4 K V 1-2:50, g5 KV 3-4:50, 205IF
Curso: CI-1201 Programación 2	Horas semanales: 4	Horario de Consulta: A convenir
Ciclo: II-2008	Modalidad: Teórico/Semestral	Ricardo Gang Vincenzi
Tipo: Requerido	Requisito: CI-1101, CI-1104	ricardogang@costarricense.cr
		Oficina 221 IF, Tel. 2202-6166

Objetivos**General**

- Proveer al estudiante la formación básica necesaria para su desempeño eficiente como programador de herramientas y como programador de aplicaciones de poca complejidad que requieran equipos de programación y el uso eficiente y efectivo de ambientes de programación orientados a objetos, cuyo uso apropiado implica la reutilización de software.

Específicos

- Comprender la diferencia entre la especificación de una clase, de una función o de un procedimiento y su implementación. O lo que es lo mismo: comprender la diferencia entre abstracción e implementación.
- Especificar y documentar clases, funciones y procedimientos
- Generar varias opciones de diseño dado un problema específico, valorarlas y escoger razonadamente una, con base en criterios tales como: simplicidad de las estructuras de datos, simplicidad de los algoritmos, ahorro evidente de memoria dinámica y de tiempo
- Comprender y aprender a usar una biblioteca de clases contenedoras y sus correspondientes iteradores, en la producción de otros componentes reutilizables y programas de aplicación de poca complejidad
- Diseñar e implementar clases contenedoras, en el contexto de una biblioteca de clases contenedoras preconstruida
- Comprender, usar e implementar algún tipo de árbol balanceado
- Comprender y aprender a usar una biblioteca de clases para la construcción de interfaces de programas de aplicación (sean estas gráficas o de texto)
- Comprender y aplicar los conceptos de la programación por eventos en el contexto del diseño e implementación de interfaces de programas de aplicación
- Comprender y aplicar los siguientes conceptos: enlace dinámico, enlace estático, sobrecarga de operadores y de funciones, polimorfismo de objetos contenedores, herencia simple, composición de clases, clases abstractas y clases concretas

Contenidos

1. **Introducción:** ¿Cómo se corre un programa?, Lenguaje->Ejecutable, Estructuras de Datos, Flujo de control (Ciclos y condicionales), Algoritmos
2. **Conceptos:** Especificación vrs Implementación, Abstracción, ADT, OOP, Documentación, Complejidad, Diseño, C++, Enlace Dinámico, Enlace Estático, Bibliotecas, Ingeniería de Software
3. **Programación Orientada a Objetos:** Historia, Conceptos básicos, Sobrecarga de operadores, Sobrecarga de funciones, Polimorfismo, Herencia, Ocultamiento de datos, Composición de clases, Clases abstractas y concretas, Plantillas
4. **Clases Contenedoras:** Arreglos, Matrices, Listas, Listas dobles, Pilas, Colas, Colas de Prioridad, Conjuntos, Bolsas, Árboles (N-Arios, AVL, 2-3, BTree, Rojo-Negro), Compuestos
5. **Hilos:** Programación concurrente, sección crítica, semáforos
6. **Interfaces:** Usabilidad, Interfaces gráficos, programación 3d
7. **Eventos:** Programación por eventos, principalmente para interfaces gráficas

Metodología

Este curso se desarrolla con la presentación magistral de conceptos teóricos y la implementación de ejercicios programados por parte del estudiante. El estudiante de este curso deberá poner en práctica conceptos adquiridos en el curso anterior de programación además de los nuevos conceptos.

Se espera que los estudiantes participen activamente en las lecciones mediante preguntas e intervenciones que enriquezcan la discusión. Se espera, además, que el estudiante ponga en práctica los conceptos del curso en las tareas programadas. En las tareas programadas el estudiante debe tomar iniciativa propia para solucionar los problemas propuestos ya que en ellas se evaluará tanto el dominio de la teoría como la capacidad de resolución de problemas.

Evaluación

Todo trabajo evaluable debe ser entregado electrónicamente al sitio del curso en el servidor de la Escuela. Esta es la única forma de entregar tareas. Solo se evaluarán los trabajos enviados a esta dirección. Los exámenes son acumulativos. La nota máxima de 10 puntos en el curso se distribuye así:

Primer examen	17%
Segundo examen	17%
Examen final	23%
Tareas programadas	19%
Tareas Cortas y Quices	13%
Trabajo en clase	11%

Para aprobar el curso el estudiante debe tener 6.75 puntos o más. Si la nota final está entre 5.75 y 6.74 tendrá derecho a realizar un examen de ampliación. En este examen el estudiante deberá obtener una nota mínima de 7.0 para aprobar el curso. En caso que obtenga una nota menor a 5.75, o de presentar el examen de ampliación con una nota inferior a 7.0 reprobará el curso.

Observaciones

- Las **tareas cortas y programadas** deberán ser entregadas el día y hora especificados en el enunciado de cada tarea al sitio (no al asistente). Generalmente la hora de entrega de cada tarea corta es el inicio de la sesión correspondiente según el siguiente cronograma. Cualquier atraso en la entrega de las mismas será sancionada de acuerdo con la siguiente regla:

A la nota obtenida se le rebajará puntos de acuerdo a la siguiente fórmula (sea i el número de días de atraso iniciando un segundo después de la fecha y hora de entrega):

$$e^{\frac{i}{\pi}} * 5$$

- El **trabajo en clase** algunas veces consistirá en un cuestionario que se entregará al inicio de la clase. Durante el transcurso de la misma se darán las respuestas a dicho cuestionario, deberá entregarse al profesor al terminar la clase.
- Las pruebas cortas (**quices**) se realizarán al menos una vez a la semana y se efectuarán en cualquier momento durante el transcurso de la lección. En los quices se evaluará la materia vista en clase hasta una semana antes (la materia es acumulativa). Dichas pruebas no se repondrán a menos que exista una justificación válida de acuerdo con los reglamentos vigentes. Las pruebas cortas serán individuales.

Fechas Importantes

Primer Examen Parcial	Setiembre 16
Segundo Examen Parcial	Octubre 24
Examen Final	Noviembre 28

Cronograma

El siguiente cronograma es tan solo una guía de los temas a ser cubiertos y todo está sujeto a cambios. Los números que aparecen en paréntesis se refieren al número de sesión en la que debe entregarse cada tarea.

Sesión 1	Agosto 12 Carta del Estudiante Introducción Conceptos	Sesión 2	Agosto 15 C++ Conceptos	Sesión 3	Agosto 19 Conceptos	Sesión 4	Agosto 22 Conceptos
TC1 (2) TeC1 Diagnóstico TP0(4)		Entrega TC1 TeC2		Quiz 1		TC2 (5)	
Sesión 5	Agosto 26 Objetos	Sesión 6	Agosto 29 Objetos	Sesión 7	Setiembre 2 Objetos	Sesión 8	Setiembre 5 Objetos
Entrega TC2 TP1 (10) Quiz 2		TeC 3		TeC 4		Quiz 3	
Sesión 9	Setiembre 9 Clases Contenedoras	Sesión 10	Setiembre 12 Clases Contenedoras	Sesión 11	Setiembre 16 Examen 1	Sesión 12	Setiembre 19 TP2
		Quiz 4 Entrega TP1 Repaso		TP 2(17)		TeC 5	
Sesión 13	Setiembre 23 Objetos	Sesión 14	Setiembre 26 Objetos	Sesión 15	Setiembre 30 Clases Contenedoras	Sesión 16	Octubre 3 Clases Contenedoras
TC3 (14)		Entrega TC3 Quiz 5		TeC 6		TeC 7	
Sesión 17	Octubre 7 TP3	Sesión 18	Octubre 10 TP3	Sesión 19	Octubre 14 Hilos	Sesión 20	Octubre 17 Hilos
Entrega TP2 TP3 (22) TC4 (19)		TeC8		Entrega TC4 Quiz 6			
Sesión 21	Octubre 21 Hilos	Sesión 22	Octubre 24 Examen 2	Sesión 23	Octubre 28 TP4	Sesión 24	Octubre 31 TP4
Quiz 7 Repaso		Entrega TP3 TP4 (31)		TC5(25)			
Sesión 25	Noviembre 4 Interfaces	Sesión 26	Noviembre 7 Interfaces	Sesión 27	Noviembre 11 Interfaces	Sesión 28	Noviembre 14 Eventos
Entrega TC5		Quiz 8		TeC9		TeC10	
Sesión 29	Noviembre 18 TC6	Sesión 30	Noviembre 21 Eventos	Sesión 31	Noviembre 25	Sesión 32	Noviembre 28
TC6 (30) TeC11		Entrega TC6		Quiz 9 Repaso		Examen 3	

Bibliografía

Principalmente se utilizará material recolectado de Internet. Como referencia, y para realizar algunas tareas, se pueden utilizar los siguientes libros y artículos.

- Aho, A., Hopcroft, J., Ullman J., "Estructuras de Datos y Algoritmos", Addison Wesley, 1988
- Aguilar, L.J., "Programación en C++, Algoritmos, estructuras de datos y objetos", Mc Graw Hill, 2000
- Deitel, H.M., Deitel, P.J., "Cómo Programar en C/C++ y Java", 4ta Edición, Prentice-Hall, 2004
- Deitel, H.M., Deitel, P.J., "How to Program JAVA™", 7ma Edición, Prentice-Hall, 2006
- Deitel, H.M., Deitel, P.J., "How to Program JAVA™", 7ma Edición, Prentice-Hall, 2006
- Di Mare, A., "Diseño y Programación Estructurada, en Fundamentos de Algoritmos y Lenguajes", Escuela de Matemática, Universidad de Costa Rica (UCR), 1978
- Di Mare, A., "Tipos Abstractos de Datos y Programación por Objetos", Reporte Técnico PIBDC-03-91, proyecto 326-89-019, Escuela de Ciencias de la Computación e Informática (ECCI), Universidad de Costa Rica (UCR), 1991
- Di Mare, A., "Reutilización de Contenedores Parametrizables con Lenguajes de Semántica Limitada", Tesis, Universidad Autónoma de Centro América, 1999
- Di Mare, A., "Una Clase Matriz Chirrisquitica Escrita en C++", Reporte Técnico ECCI-2004-02, Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica, 2004
- Di Mare, A., "Una clase C++ completa para conocer y usar árboles", Reporte Técnico ECCI-2004-03, Escuela de Ciencias de la Computación e Informática, Universidad de Costa Rica, 2004
- Stroustrup, B., "The C++ Programming Language", 3ra Edición, Addison Wesley, 1998