



UNIVERSIDAD DE COSTA RICA  
RECINTO DE TACARES  
INFORMATICA EMPRESARIAL  
CURSO: IF-2000  
PROGRAMACION I  
PROF: MARCO ACOSTA  
II-2007

## DESCRIPCIÓN DEL CURSO

Este curso pretende introducir al estudiante en el campo de la elaboración de programas computacionales. Se presentan los componentes básicos de todo lenguaje de programación y el manejo dinámico de memoria. Se hace énfasis en que el estudiante asimile una serie de conceptos básicos relacionados con la calidad del software. Una característica importante del curso es que se introduce al estudiante de manera temprana a la programación orientada a objetos.

## OBJETIVO GENERAL

Iniciar al estudiante en la elaboración de solución de problemas a través de un lenguaje de programación.

## OBJETIVOS ESPECÍFICOS

Este curso facilita al estudiante la información y las herramientas necesarias para que pueda:

1. Determinar las características de un buen producto de software
2. Entender la estructura y funcionalidad de un lenguaje de programación
3. Apreciar las ventajas que ofrece el diseño de programas.
4. Comprender la conveniencia de separar el código de los programas en módulos independientes.

## CONTENIDO DEL CURSO

- |      |  |       |   |
|------|--|-------|---|
| 1.   | Caracterización de un buen software                          | 4.2.  | Uso de librerías  |
| 1.1. | Confiabilidad  | 4.3.  | Importancia.  |
| 1.2. | Claridad   | 4.4.  | Interfaces  |
| 1.3. | Relación costo-beneficio                                     | 4.5.  | Compilación separada  |
| 1.4. | Adaptabilidad del producto                                   | 4.6.  | Enlace de los módulos orientada a objetos   |
| 1.5. | Reutilización  | 4.7.  | Constructores y destructores  |
| 1.6. | Estándares de documentación (interna y externa)              | 4.8.  | Envío de mensajes   |
|      |  | 4.9.  | Polimorfismos   |
|      |  | 4.10. | Sobrecarga de operadores y métodos  |
|      |  | 4.11. | Herencia y plantillas.  |
| 2.   | Principios de la ingeniería de software                      | 5.    | Arreglos y vectores   |
| 2.1. | Uso de la abstracción  | 5.1.  | Análisis de vectores y operaciones sobre ellos                                      |
| 2.2. | Modularidad  | 5.2.  | Algoritmos para clasificación y búsqueda básica                                     |
| 2.3. | Estructuras jerárquicas                                      | 6.    | Manejo dinámico de memoria.   |
| 2.4. | Ocultamiento de la información.                              | 6.1   | Concepto de puntero   |
| 3.   | Objetos y tipos  | 6.2   | Ventajas y desventajas de las estructuras dinámicas vrs las estáticas.              |
| 3.1. | Concepto de objetos simples (variables)                      | 6.3   | Tipos de definidos por el programador.  |
| 3.2. | Características de un buen identificador para el objeto.     | 6.4   | Funciones que realizan las asignación de memoria y el trabajo con objetos dinámicos |
| 3.3. | Palabras reservadas  | 7     | Archivos y registros  |
| 3.4. | Tipos de datos básicos (enteros, caracteres, números reales) | 7.1.  | Registros   |
| 4.   | Programación orientada a objetos                             | 7.2.  | Archivos, tipos y operaciones básicas.  |
| 4.1. | Clases y objetos   | 8.    | Recursividad.   |

## METODOLOGÍA

Se debe fomentar el trabajo individual y algunos en grupo, enfrentar al estudiante con problemas reales para que les encuentre una solución programada. Con lo que se pretende que el estudiante desarrolle más su iniciativa e inventiva al resolver problemas. Se abordarán los temas mediante clases magistrales, ejercicios prácticos, presentaciones, desarrollo de programas en laboratorio y tareas programadas.

Las Tareas Programadas reforzarán los principios estudiados en clase. En este aspecto es muy importante considerar por parte de los estudiantes, el cumplir cabalmente con los requerimientos que el profesor defina para la elaboración de la tarea, los cuales serán entregados por el profesor en su debido tiempo. Se harán pruebas cortas en cualquier momento.

Todas las Tareas Programadas deberán entregarse en la fecha programada, en caso contrario se rebajará un 50% del total de la nota obtenida por un día de atraso, al segundo día de haber pasado la fecha de entrega **NO** se recibirán ninguna investigación o

tarea. Asimismo deberán entregar la *documentación* respectiva y brindar una *demonstración (en caso de tarea programada)*, con valor de 70% y 30% respectivamente.

La documentación incluye: 1) PORTADA: 1. Nombre y sigla del curso, 2. Nombre, número de carnet y nota interna de los estudiantes si es en grupo, 3. Número de Tarea Programada y breve descripción del problema; 2) DESARROLLO: 1. Índice de contenido, 2. Objetivos generales y específicos, 3. Descripción detallada del problema, asimismo información recolectada según la Bibliografía consultada, 4. Algoritmo y listado fuente, 5. Diseño gráfico de la Base de Datos (si hubiera), 6. Diseño de la Interfaz, 7. Salidas o Resultados; 3) PARTE FINAL: 1. Conclusiones, 2. En caso de que no le salió explicar las razones técnicas de ello. 3. Bibliografía que incluya las direcciones Internet consultadas.

#### **EVALUACION**

<b>Rubro</b>	<b>Porcentaje</b>
Exámenes	40
Pruebas Cortas	20
Tareas Programadas	40

#### **BIBLIOGRAFÍA**

- 1.Osvaldo Cairó. Metodología de la Programación. Algoritmos, Diagramas de Flujo y Programas. Tomo I. AlfaOmega Grupo Editor, S.A. de C.V. 1995.
- 2.Osvaldo Cairó. Metodología de la Programación. Algoritmos, Diagramas de Flujo y Programas. Tomo II. AlfaOmega Grupo Editor, S.A. de C.V. 1996.
- 3.Kris Jamsa. Aprenda C++ Paso a Paso. . AlfaOmega Grupo Editor, S.A. de C.V. 1996.
- 4.Chris H. Pappasa, William H. Murray, III. Manual de Borland C++. Osborne McGraw-Hill. 1993.
- 5.Byron S. Gottfried. Programación en Pascal. Teoría y 391 Problemas Resueltos 35 Ejemplos Completos de Programación.Serie Chaum en Computación. 1986.
- 6.Deitel y Deitel, "JAVA Cómo Programar", Edit. Prentice Hall 5ta Ed.
- 7.Deitel (Harvey) & Deitel (Paul), "Cómo programar en C++", Cuarta edición, Prentice Hall, 2003.