



PROGRAMA CURSO: PROGRAMACIÓN I
II Semestre, 2012

Datos Generales

Sigla: IF-2000

Nombre del curso: Programación I

Tipo de curso: Teórico-Práctico

Número de créditos: 4

Número de horas semanales presenciales: 8

Número de horas semanales de trabajo independiente del estudiante: mínimo 6

Requisitos: IF-1300 Introducción a la computación e informática

Correquisitos: No tiene

Ubicación en el plan de estudio: II Ciclo

Horario del curso: K 16:00 a 20:00 y M 13:00 a 16:50

Suficiencia: No

Tutoría:No

Datos del Profesor

Nombre: Mci. Francisco Blanco Chavarría

Correo Electrónico: francisco.blanco@ucrsi.info

Horario de Consulta: V 13:00 a 17:00

1. 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.

2. Objetivo General

Introducir al estudiante en el análisis de problemas y la abstracción de soluciones que permitan la construcción de programas a través de un lenguaje de programación.

3. Objetivos específicos

Al finalizar el curso el o la estudiante estará en capacidad de:



- a. Identificar los atributos que caracterizan un producto de software de calidad para su aplicación en el desarrollo de programas.
- b. Describir la estructura de un lenguaje de programación para comprender su funcionalidad.
- c. Reconocer las ventajas que ofrece el diseño en la creación de programas de software.
- d. Comprender la conveniencia de separar el código de los programas en módulos independientes.
- e. Estudiar el concepto de programación orientada a objetos así como mostrar la importancia y aplicación del mismo en la actualidad

1. Contenidos

- 1 Caracterización de un buen software
 - 1.1 Conceptos y ejemplos de
 - 1.1.1 Confiabilidad
 - 1.1.2 Claridad
 - 1.1.3 Relación costo-beneficio
 - 1.1.4 Adaptabilidad del producto
 - 1.1.5 Reutilización
 - 1.1.6 Estándares de documentación (interna y externa)
- 2 Principios de la ingeniería de software
 - 2.1 Concepto, funcionamiento, ejemplo de
 - 2.1.1 Uso de la abstracción
 - 2.1.2 Modularidad
 - 2.1.3 Encapsulamiento
 - 2.1.4 Ocultamiento de la información
- 3 Programación orientada a objetos (POO)
 - 3.1 Concepto, funcionamiento, ejemplo de
 - 3.1.1 Variables y métodos estáticos.
 - 3.1.2 Atributos tipo Objeto (Composición/Agregación)
 - 3.1.3 Sobrecarga de métodos
 - 3.1.4 Encadenamiento de métodos
 - 3.1.5 Herencia
 - 3.1.5.1 Superclase, subclase
 - 3.1.5.2 Clase abstracta
 - 3.1.5.3 Clase interface
 - 3.1.6 Polimorfismo
 - 3.1.6.1 Concepto e importancia del polimorfismo
 - 3.1.6.2 Aplicación del polimorfismo en la POO
 - 3.1.7 Uso de bibliotecas
 - 3.1.7.1 Interfaz Gráfica de usuario
- 4 Arreglos y matrices
 - 4.1 Arreglos o Vectores (arreglos unidimensionales)
 - 4.1.1 Concepto, importancia, aplicación de los vectores
 - 4.1.2 Análisis de vectores y operaciones sobre ellos
 - 4.1.2.1 Crear, modificar, insertar y eliminar un vector



- 4.1.2.2 Operaciones sobre vectores (multiplicación, suma, promedio, mínimo, máximo, etc.)
- 4.2 Matrices (arreglos bidimensionales)
 - 4.2.1 Concepto, importancia, aplicación de las matrices
 - 4.2.2 Análisis de matrices y operaciones sobre ellas
 - 4.2.2.1 Crear, modificar, insertar y eliminar una matriz
 - 4.2.2.2 Operaciones sobre matriz (recorridos de fila y columna, diagonales, etc.)
- 4.3 Algoritmos para clasificación y búsqueda básica en vectores
 - 4.3.1 Ordenamiento en vectores
 - 4.3.1.1 Algoritmo Burbuja
 - 4.3.2 Búsqueda en vectores
 - 4.3.2.1 Algoritmo búsqueda Secuencial
 - 4.3.2.2 Algoritmo búsqueda Binaria
- 5 Manejo de memoria
 - 5.1 Concepto de puntero, importancia, aplicación, visualización en un mapa de memoria
 - 5.2 Ventajas y desventajas de las estructuras dinámicas vrs las estáticas.
- 6 Archivos y registros
 - 6.1 Concepto archivo y registro, relación entre archivo y registro
 - 6.2 Importancia y aplicación de los archivos
 - 6.3 Tipos de archivos
 - 6.3.1 Secuenciales (texto y binarios)
 - 6.3.2 Aleatorios
 - 6.4 Operaciones básicas sobre archivos.
 - 6.4.1 Crear, abrir, modificar, cerrar un archivo.
 - 6.4.2 Búsqueda de datos en archivos.
- 7 Recursividad.
 - 7.1 Concepto e importancia
 - 7.2 Diferencia de recursividad vrs. ciclos.
 - 7.3 Ejemplos básicos de recursivos.

1. 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 de investigación..



2. Evaluación

Descripción	Porcentaje
Examen Parcial I	15%
Examen Parcial II	15%
Examen Parcial III	15%
Quices, Tareas Cortas , otros	10%
Laboratorios	15%
Tarea Programada 1	15%
Tarea Programada 2	15%
Total:	100%

Consideraciones sobre la evaluación

- Los Quices se harán sin previo aviso en cualquier momento de la lección y no se harán reposiciones (excepto por las disposiciones establecidas en el reglamento de Régimen Académico Estudiantil)
- Las fechas de las evaluaciones del cronograma están sujetas a cambio dependiendo del avance de los temas vistos.
- Los rubros de calificación de cada una de las evaluaciones serán entregadas junto con la especificación de la misma.
- Las tareas programadas se realizarán en grupos establecidos por el profesor(a).
- Para las tareas programadas se realizará una comprobación individual para determinar la participación de los integrantes.
- Las tareas programadas deben ser entregadas a la hora y fecha indicadas en el enunciado del proyecto. En caso de que se atrase un día, perderán el 40% del valor del trabajo.
- Para los demás aspectos de evaluación no se aceptarán entregas después de la fecha y hora solicitada.
- Mantener celulares en modo silencioso o apagado durante las lecciones.
- La aparición de un celular durante un examen u otra prueba anulará automáticamente la misma.
- La comprobación de que cualquier evaluación, tarea, laboratorio, proyecto o examen es una copia hará que se apliquen las sanciones que contemple la reglamentación universitaria.
- Se utilizará la plataforma campus.portapapeles.net como apoyo del curso, entrega de material del curso por parte del profesor(a) y para recibo y calificación de algunas de las evaluaciones.
- Para comunicación con el profesor(a) se utilizará únicamente el correo de la carrera (@ucrsi.info) por lo cual el estudiante debe tenerlo activo y revisarlo continuamente.



1. Cronograma.

	SEMANA	TEMA
1	06 – 12 Agosto	Presentación curso y entrega carta al estudiante Repaso POO
2	13 – 19 Agosto	Repaso POO Tema 1: Caracterización de un buen software Tema 2: Principios de la ingeniería de software Tema 3: Programación orientada a objetos
3	20 – 26 Agosto	Tema 3: Programación orientada a objetos
4	27 – 02 Septiembre	Tema 3: Programación orientada a objetos
5	03 – 09 Septiembre	I Examen Tema 3: Programación orientada a objetos (3.1.6 Polimorfismo)
6	10– 16 Septiembre	Tema 3: Programación orientada a objetos
7	17 – 23 Septiembre	Tema 4: Arreglos y matrices
8	24 – 30 Septiembre	Tema 4: Arreglos y matrices
9	01 – 07 Octubre	Entrega I Tarea Programada Comprobación I Tarea Programada Tema 4: Arreglos y matrices
10	08 – 14 Octubre	Tema 4: Arreglos y matrices II Examen
11	15 – 21 Octubre	Tema 5: Manejo de memoria
12	22 – 28 Octubre	Tema 6: Archivos y registros
13	29 – 04 Noviembre	Tema 6: Archivos y registros
14	05 – 11 Noviembre	Tema 6: Archivos y registros Tema 8: Recursividad
15	12 – 18 Noviembre	Tema 8: Recursividad
16	19 – 25 Noviembre	III Examen
17	26 – 02 Diciembre	Entrega II Tarea Programada Comprobación II Tarea Programada Entrega de promedios
18	03 – 09 Diciembre	Ampliación

Algunos de los siguientes temas se pueden investigar en tareas cortas como complemento a los contenidos: documentación javadoc, excepciones, genéricos, pruebas con JUnit



1 Bibliografía

- Deitel, P. J., & Deitel, H. M. (2008). *Cómo programar en Java* (Séptima ed.). México: Pearson Education.
 - Wu, C. T. (2008). *Programación en Java*. México: McGraw-Hill.
 - Joyanes Aguilar, L. (2008). *Fundamentos de programación* (Cuarta ed.). México: MacGraw-Hill.
 - Carballo, F. J. (2008). *Java 2, Curso de programación* (Tercera ed.). México: Alfaomega.
 - Ceballos, F. J. (2008). *Java 2, Interfaces gráficas y aplicaciones para Internet* (Tercera ed.). México: Alfaomega.
 - Jamsa, K. (1996). *Aprenda C++ Paso a Paso*. AlfaOmega.
 - Sánchez Allende, J., & otros. (2001). *Java 2*. McGraw-Hil.
-