



PROGRAMA CURSO: PROGRAMACIÓN I
II Semestre, 2013

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 17:00 a 20:50 y M 17:00 a 20:50

Suficiencia:

Tutoría:

Datos del Profesor

Nombre: Kenneth Cruz Arias

Correo Electrónico: Kenneth_Cruz@ucrso.info

Horario de Consulta: L 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

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

3.2 Arreglos y matrices

3.2.1 Arreglos o Vectores (arreglos unidimensionales)

- 3.2.1.1 Concepto, importancia, aplicación de los vectores
- 3.2.1.2 Análisis de vectores y operaciones sobre ellos
 - 3.2.1.2.1 crear, modificar, insertar y eliminar un vector
 - 3.2.1.2.2 multiplicación, suma, promedio, mínimo, máximo, etc.
- 3.2.1.3 Algoritmos para clasificación y búsqueda básica en vectores
 - 3.2.1.3.1 Ordenamiento en vectores
 - 3.2.1.3.1.1 Algoritmo Burbuja
 - 3.2.1.3.2 Búsqueda en vectores
 - 3.2.1.3.2.1 Algoritmo búsqueda Secuencial
 - 3.2.1.3.2.2 Algoritmo búsqueda Binaria

3.2.2 Matrices (arreglos bidimensionales)

- 3.2.2.1 Concepto, importancia, aplicación de las matrices
- 3.2.2.2 Análisis de matrices y operaciones sobre ellas
 - 3.2.2.2.1 crear, modificar, insertar y eliminar
 - 3.2.2.2.2 recorridos de filas, columnas, diagonales, etc.

3.3 Herencia

- 3.3.1 Superclase, subclase



- 3.3.2 Clase abstracta
- 3.3.3 Clase interface
- 3.4 Polimorfismo
 - 3.4.1 Concepto e importancia del polimorfismo
 - 3.4.2 Aplicación del polimorfismo en la POO
- 3.5 Uso de bibliotecas
 - 3.5.1 Interfaz Gráfica de Usuario (GUI)
- 4 Modelado de sistemas orientados a objetos con UML
 - 4.1 El lenguaje de modelado unificado (UML), importancia y utilización.
 - 4.2 Diagramas de Clase: Identificación de Clases (atributos, operaciones), herencia e interfaces.
 - 4.3 Relaciones de clases: dependencia, generalización, asociación (nombre, rol, multiplicidad, navegación, agregación y composición).
- 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.

Los siguientes temas serán evaluados en el rubro de exposiciones: archivos(todos los tipos), excepciones, genéricos

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	18%
Examen Parcial II	18%
Examen Parcial III	18%
Quices, Tareas Cortas , otros	5%
Exposición	5%
Laboratorios	12%
Tarea Programada 1	12%
Tarea Programada 2	12%
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 su especificación.
- 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, de la cual dependerá el total de la nota.
- 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 (excepto por las disposiciones establecidas en el reglamento de Régimen Académico Estudiantil)
- Para cualquier ejercicio programado (independientemente del rubro de evaluación al que pertenezca), es responsabilidad del estudiante entregar una versión de la aplicación que compile sin errores, si el resultado entregado no cumple con esta característica el ejercicio será calificado con nota 0.
- Los temas de investigación también serán evaluados dentro de rubro los exámenes.
- 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.
- 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, ya que es el medio que se utilizará para avisarle sobre actualizaciones en el sitio digital del curso, avisos, asignación de tareas, etc.
- Se utilizará la plataforma de Moodle de Informática Empresarial como apoyo del curso, entrega de material del curso por parte del profesor(a) y para recibo y calificación de algunas de las evaluaciones.

1. Cronograma.

	SEMANA	TEMA
1	12 – 18 Agosto	Presentación curso y entrega carta al estudiante Repaso POO
2	19 – 25 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	26 – 01 Septiembre	Tema 3: Programación orientada a objetos
4	02 – 08 Septiembre	Tema 3: Programación orientada a objetos
5	09 – 15 Septiembre	Tema 3: Programación orientada a objetos
6	16– 22 Septiembre	I Examen Tema 3: Programación orientada a objetos (3.3 Herencia)
7	23 – 29 Septiembre	Tema 3: Programación orientada a objetos
8	30 – 06 Octubre	Tema 4: UML
9	07 – 13 Octubre	Tema 5: Manejo de memoria Entrega I Tarea Programada Comprobación I Tarea Programada
10	14 – 20 Octubre	II Examen Exposiciones
11	21 – 27 Octubre	Exposiciones
12	28 – 03 Noviembre	Exposiciones
13	04 – 10 Noviembre	Exposiciones
14	11 – 17 Noviembre	Exposiciones Tema 8: Recursividad
15	18 – 24 Noviembre	Tema 8: Recursividad
16	25 – 01 Diciembre	III Examen
17	02 – 08 Diciembre	Entrega II Tarea Programada Comprobación II Tarea Programada Entrega de promedios
18	09 – 15 Diciembre	Ampliación

1 Bibliografía

- Deitel, P. J., & Deitel, H. M. (2008). *Cómo programar en Java (Séptima ed.)*. México: Pearson Education.
- Deitel, P. J., & Deitel, H. M. (2012). *Cómo programar en Java (Novena ed.)*. México: Pearson Education.
- Wu, C. T. (2008). *Programación en Java*. México: McGraw-Hill.
- Booch, G., Jacobson, I., & Rumbaugh, J. (2006). *El lenguaje de modelado unificado, UML*. Pearson Education.
- Fowler, M. (1997). *UML Gota a Gota*. México: Pearson Education.
- Schach, S. (2005). *Análisis y Diseño Orientado a Objetos con UML y el proceso unificado*. México: Mc Graw Hill.
- Schmuller, J. (2001). *Aprendiendo UML en 24 horas*. México: Pearson Education.



- 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.