



PROGRAMA CURSO: PROGRAMACIÓN I
II Semestre, 2016

Datos Generales

Sigla: IF2000

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: 12

Requisitos: IF1300 Introducción a la computación e informática

Correquisitos: Ninguno

Ubicación en el plan de estudio: II Ciclo

Horario del curso: K 5:00 pm a 9:00 y V 5:00 a 9:00

Suficiencia: No

Tutoría: No

Datos del Profesor

Nombre: Mci. Mario Alberto Barrantes Quesada

Correo Electrónico: mario.barrantes@ucrsi.info

Horario de Consulta: M 5:00 pm a 9:00 pm

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:

- Identificar los atributos que caracterizan un producto de software de calidad para su aplicación en el desarrollo de programas.
- 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

4. 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** 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, ejemplos de
 - 3.1.1** Clases y Objetos
 - 3.1.2** Métodos: de instancia, constructores, destructores, set y get, llamado de métodos entre clases y en la misma clase (modularidad), paso por valor,
 - 3.1.3** Conceptos y diferenciación de referencia e instancia
 - 3.1.4** Variables y métodos de clase (estáticos).
 - 3.1.5** Relación entre clases (Composición/Agregación)
 - 3.1.6** Sobrecarga de métodos
 - 3.1.7** Encadenamiento de métodos
 - 3.2** Arreglos
 - 3.2.1** 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 Concepto, importancia y aplicación de herencia en POO
- 3.3.2 Superclase, subclase
- 3.3.3 Clase abstracta
- 3.3.4 Interfaces
- 3.4 Polimorfismo
 - 3.4.1 Concepto, importancia y aplicación del polimorfismo en POO
- 3.5 Librerías
 - 3.5.1 Uso y creación de librerías
 - 3.5.2 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, importancia y aplicación de la recursividad
 - 7.2 Diferencia de recursividad vrs. ciclos.

Los siguientes temas podrían ser evaluados en el rubro de exposiciones: archivos(todos los tipos), excepciones, genéricos y temas introductorios a los cursos IF3000 e IF3001.

5. 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.



6. Evaluación

Descripción	Porcentaje
Examen Parcial I	25%
Examen Parcial II	25%
Pruebas cortas, Tareas Cortas , otros	10%
Exposición	5%
Laboratorios	10%
Proyecto Programado	25%

Total: 100%

Consideraciones sobre la evaluación

- Las pruebas cortas se harán sin previo aviso en cualquier momento de la lección. Dichas pruebas podrán ser de comprobación de tareas cortas, tareas programadas, laboratorios, exposiciones, comprobaciones de lectura, comprobaciones de la materia vista durante la clase o en clases anteriores.
- En las clases se realizarán diferentes ejercicios de programación y otras prácticas relacionadas con los temas del curso, durante las clases se podrán realizar laboratorios los cuales consisten en prácticas a realizar durante la clase y entregadas al final de la misma, a no ser que el profesor indique una fecha y hora posterior para la entrega. Los laboratorios también pueden ser realizados sin previo aviso.
- El uso del laboratorio para fines no académicos está prohibido.
- El valor de cada prueba corta y laboratorio depende de la cantidad de pruebas que se realicen durante el curso.
- Las fechas de las evaluaciones del cronograma están sujetas a cambio dependiendo del avance de los temas vistos.
- Para ningún tipo de evaluación se aceptarán entregas después de la fecha y hora solicitadas en el sitio del curso o entregados por otros medios. Es responsabilidad de cada estudiante o grupo de trabajo enviar el archivo con suficiente tiempo antes de la hora límite.
- Los trabajos en los que deban crear un programa de software deberán ser entregados con el código fuente completo y en el formato adecuado para poder ser revisado, compilado y ejecutado. Si un programa tiene errores de compilación o algún otro tipo de error que impida su ejecución será calificado con nota CERO.
- Sólo se reponen evaluaciones según lo estipulado en el artículo 24 del Reglamento de Régimen Académico de la UCR y utilizando el procedimiento ahí descrito.
- Mantener celulares en modo silencioso o apagado durante las lecciones.
- Durante las pruebas individuales (exámenes, pruebas cortas y otras indicadas por el profesor), los celulares u otros medios de comunicación deberán estar apagados y guardados en el bulto del estudiante. La aparición de un celular durante un examen u otra prueba anulará automáticamente la misma por lo que tendrá una nota de CERO.
- La comprobación de que cualquier evaluación presenta copia o algún tipo de fraude, total o parcial, se calificará con nota de CERO para TODOS los involucrados y se seguirá el trámite académico respectivo que contempla la reglamentación universitaria sin previo aviso.
- 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 evaluaciones.



7. Cronograma.

	SEMANA	TEMA
1	08 – 14 Agosto	Presentación y entrega del programa del curso Repaso Introducción Tema 1: Caracterización de un buen software Tema 2: Principios de la ingeniería de software
2	15 – 21 Agosto	Tema 3: Programación orientada a objetos
3	22 – 28 Agosto	Tema 3: Programación orientada a objetos
4	29 – 04 Setiembre	Tema 3: Programación orientada a objetos
5	05 – 11 Setiembre	I Examen Tema 3: Programación orientada a objetos (Herencia)
6	12 – 18 Setiembre	Tema 3: Programación orientada a objetos
7	19 – 25 Setiembre	Tema 3: Programación orientada a objetos
8	26 – 02 Octubre	Tema 3: Programación orientada a objetos Tema 4: UML
9	03– 09 Octubre	Tema 5: Manejo de memoria Tema 7: Recursividad Entrega y defensa de I parte Proyecto Programado
10	10 – 16 Octubre	Tema 7: Recursividad
11	17 – 23 Octubre	Tema 7: Recursividad II Examen
12	24 – 30 Octubre	27 y 28 de octubre IV Foro de Informática Empresarial Exposiciones
13	31 – 06 Noviembre	Exposiciones
14	07 – 13 Noviembre	Exposiciones
15	14 – 20 Noviembre	Exposiciones
16	21 – 27 Noviembre	Exposiciones Entrega y defensa de II parte Proyecto Programado
17	28 – 04 Diciembre	Entrega de promedios
18	05 – 11 Diciembre	Ampliación

8 Bibliografía

- 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.
- Schach, S. (2005). *Análisis y Diseño Orientado a Objetos con UML y el proceso unificado*. México: Mc Graw 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.