

PROGRAMA CURSO: PROGRAMACIÓN I
II Semestre, 2018

Datos Generales

Sigla del curso: IF2000
Nombre del curso: Programación I
Tipo de curso: Teórico-Práctico
Créditos: 4
Horas lectivas semanales: 8
Requisitos: IF1300 Introducción a la computación e informática
Correquisitos: Ninguno
Ubicación en el plan de estudio: II Ciclo
Horario del curso: V 16:00 a 19:50 y S 8:00 a 11:50
Tutoría: No
Suficiencia: No
Modalidad: Presencial

Datos del Profesor

Nombre: Lic. Elliot Manrique Castro Castro
Correo Electrónico: elliot.castro@ucrsi.info, manriquec@gmail.com
Horario de consulta presencial: L 18:00 a 22:00 Handguts(manriquec@gmail.com)

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

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.

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.
- Reconocer las ventajas que ofrece el diseño en la creación de programas de software.
- Comprender la conveniencia de separar el código de los programas en módulos independientes.
- Estudiar el concepto de programación orientada a objetos así como mostrar la importancia y aplicación del mismo en la actualidad

Contenidos

- 1 Caracterización de un buen software
 - 1.1 Confiabilidad
 - 1.2 Claridad
 - 1.3 Relación costo-beneficio
 - 1.4 Adaptabilidad del producto
 - 1.5 Reutilización
 - 1.6 Estándares de documentación (interna y externa)
- 2 Principios de la ingeniería de software
 - 2.1 Abstracción
 - 2.2 Modularidad
 - 2.3 Encapsulamiento
 - 2.4 Ocultamiento de la información
- 3 Objetos y tipos
 - 3.1 Concepto de objetos simples (variables)
 - 3.2 Características de un buen identificador para el objeto.
 - 3.3 Palabras reservadas
 - 3.4 Tipos de datos básicos (enteros, caracteres, números reales)
- 4 Programación orientada a objetos
 - 4.1 Clases y objetos
 - 4.2 Uso de librerías
 - 4.3 Importancia.
 - 4.4 Interfaces
 - 4.5 Compilación separada
 - 4.6 Enlace de los módulos orientada a objetos
 - 4.7 Constructores y destructores
 - 4.8 Envío de mensajes

- 4.9 Sobrecarga de métodos
- 4.10 Herencia y plantillas
- 4.11 Polimorfismo
- 5 Arreglos y vectores
 - 5.1 Análisis de vectores y operaciones sobre ellos
 - 5.2 Algoritmos para clasificación y búsqueda básica
- 6 Manejo dinámico de memoria
 - 6.1 Concepto de puntero
 - 6.2 Ventajas y desventajas de las estructuras dinámicas vrs las estáticas
 - 6.3 Tipos definidos por el programador
 - 6.4 Funciones que realizan la asignación de memoria y el trabajo con objetos dinámicos
- 7 Archivos y registros
 - 7.1 Registros
 - 7.2 Archivos, tipos y operaciones básicas
- 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 de investigación.

Evaluación

Descripción	Porcentaje
Examen Parcial I	22%
Examen Parcial II	23%
Laboratorios	12%
Pruebas cortas, Tareas Cortas, otros	10%
Tarea Programada 1	15%
Tarea Programada 2	18%

Total:100%

Consideraciones sobre la evaluación

- La plataforma oficial que se utilizará en el curso es la plataforma institucional de Mediación Virtual (<https://mediacionvirtual.ucr.ac.cr>). La entrega de tareas, proyectos y otros evaluables será mediante el aula virtual del curso, el día y la hora indicados

previamente por el docente. No se recibirán por ningún otro medio ni fuera de la fecha y hora establecida.

- Para comunicación entre profesores y estudiantes se utilizarán únicamente las cuentas de correo de la universidad (@ucr.ac.cr) y/o de la carrera (@ucrso.info) por lo cual el estudiante debe tenerlos activos y revisarlos continuamente, ya que es el medio por el cual se les informará sobre actualizaciones en el aula virtual, evaluaciones, asignación de tareas, entre otros.
- Durante las lecciones debe mantener los celulares en modo silencioso.
- Durante las evaluaciones el uso de teléfonos celulares, tabletas o cualquier otro dispositivo de comunicación está totalmente prohibido dentro y fuera del aula mientras el estudiante no haya hecho entrega de su evaluación. Todo dispositivo que traiga consigo deberá permanecer apagado y guardado en su bolso o bulto. El uso o aparición de algún dispositivo anulará automáticamente la evaluación de todos los involucrados.
- Los rubros de calificación de cada evaluación serán entregados junto con la especificación de la misma
- Toda evaluación será comunicada al estudiante al menos 5 días hábiles antes de realizarse, a excepción de las pruebas cortas o “quices”, según lo establece el Reglamento de Régimen académico.
- Las evaluaciones deben realizarse únicamente dentro de las instalaciones de la sede o recinto donde se imparte el estudiante, y no en otro lugar aunque sea parte de la UCR.
- En caso de ausencia a alguna evaluación se procederá de acuerdo a lo establecido en el Artículo 24 del Reglamento de Régimen Académico Estudiantil de la UCR para su reposición.
- La detección de una posible copia o plagio, total o parcial, en cualquier evaluación se anulará automáticamente a los involucrados y se elevará el caso a la Dirección de la Sede de Occidente para que se aplique el proceso y las sanciones correspondientes según el Reglamento de Orden y Disciplina Estudiantil.
- Todo trabajo escrito deberá entregarse en formato de documento abierto (odf) o en formato de documento portátil (pdf).
- Para todas las tareas o investigaciones solamente se permite el uso de sitios web de corte académico, como por ejemplo artículos o libros de revistas indexadas, enciclopedias Web, sitios de noticias, así como el uso de libros y periódicos. No se permite utilizar material de Internet de fuentes de información que se pueden corroborar o sean poco confiables, de lo contrario no se tomará como bibliografía o referencia válida y la nota de la evaluación será cero.
- Dentro del aula/laboratorio sólo pueden permanecer estudiantes matriculados en el curso y grupo, no se permiten “oyentes”, estudiantes matriculados en otros grupos, administrativos o terceras personas salvo que hayan sido invitados explícitamente por el profesor para fines del curso.
- El uso del aula y/o del laboratorio es exclusivo para fines académicos propios del curso. Cualquier otro uso está prohibido.
- Durante la clase velar por el orden y aseo, principalmente cuando se dan lecciones en los laboratorios, está prohibido consumir alimentos dentro de los laboratorios y no deben dejar basura de ningún tipo al terminar la clase.

- También asegurar que todos los equipos queden apagados al igual que los aires acondicionados y los pupitres ordenados.
- El examen de ampliación contendrá todos los contenidos del curso.
- Los laboratorios, investigaciones, exposiciones, tareas programadas y cualquier asignación grupal se realizarán en grupos que el profesor definirá, según la cantidad de estudiantes del curso.
- Para las evaluaciones que impliquen programación se realizará una comprobación individual para determinar la participación de los integrantes, de la cual dependerá el total de la nota.
- Si algún estudiante es expulsado(a) de su grupo de trabajo debido a incumplimiento de labores justificadas adecuadamente, perderá los puntos respectivos y no podrá continuar con el trabajo. Las pruebas respectivas para expulsar a un compañero(a) deben ser presentadas por escrito y con copia al o a los involucrados y firmadas por los miembros, donde se exponga la situación presentada.
- La no presentación de código fuente en tareas o proyectos programados, se calificará con nota cero. Aquellos programas que no cumplan con los requerimientos mínimos establecidos en el enunciado o que presenten errores de compilación se calificará de igual manera con nota cero.
- El valor de cada prueba corta, tarea y/o laboratorio depende de la cantidad total que se realicen durante el curso, sacando un promedio de las notas obtenidas para obtener el porcentaje correspondiente.
- Las fechas de las evaluaciones del cronograma están sujetas a cambio, previamente comentados en clase, dependiendo del avance de los temas en el cronograma.
- Las pruebas cortas podrán contemplar temas abarcados en tareas cortas, tareas programadas, laboratorios, exposiciones, lecturas, presentaciones y también cualquier materia vista durante las clases.
- Los laboratorios se deben de realizar en clase, excepto que se indique lo contrario. Si un estudiante no firma la hoja de entrega del laboratorio en clase no se revisará el laboratorio aunque lo haya entregado en el aula virtual.

Cronograma

	SEMANA	TEMA
1	13 – 19 Agosto	Presentación y entrega del programa del curso Diagnóstico Módulo 1 - Programación Orientada a Objetos, Primera Parte: <ul style="list-style-type: none"> • Clases y Objetos • Caracterización de un buen software y principios de POO • Métodos: de instancia, constructores, destructores, set y get, llamado de métodos entre clases y en la misma clase (modularidad), paso por valor. • Conceptos y diferenciación de referencia e instancia

		<ul style="list-style-type: none"> • Variables y métodos de clase (estáticos). • Relación entre clases (Composición/Agregación) • Sobrecarga de métodos • Encadenamiento de métodos • Concepto de puntero, importancia, aplicación, visualización en un mapa de memoria • UML
2	20 – 26 Agosto	<p>Módulo 1 - Continuación</p> <p>Módulo 2 – Arreglos:</p> <ul style="list-style-type: none"> • Concepto, funcionamiento y algoritmos de vectores • Concepto, funcionamiento y algoritmos de matrices
3	27 – 02 Setiembre	Módulo 2 - Continuación
4	03 – 09 Setiembre	Módulo 2 - Continuación
5	10 – 16 Setiembre	Módulo 2 - Continuación
6	17 – 23 Setiembre	Módulo 2 - Continuación
7	24 – 30 Setiembre	<p>I Examen</p> <p>Módulo 3 - Programación Orientada a Objetos, Segunda Parte:</p> <ul style="list-style-type: none"> • Herencia y Polimorfismo en POO: concepto y aplicación de herencia, superclases, subclasses, clases abstractas, polimorfismo. • Uso y creación de librerías • Interfaz Gráfica de Usuario (GUI)
8	01 – 07 Octubre	Módulo 3 - Continuación
9	08 – 14 Octubre	<p>Módulo 3 - Continuación</p> <p>Entrega y defensa de I Tarea Programada</p>
10	15 – 21 Octubre	Módulo 3 - Continuación
11	22 – 28 Octubre	<p>Módulo 4 – Recursividad</p> <ul style="list-style-type: none"> • Concepto y aplicación de la recursividad • Comparación de recursividad vrs ciclos.
12	29 – 04 Noviembre	Módulo 4 - Continuación
13	05 – 11 Noviembre	<p>II Examen</p> <p>Módulo 5 – Archivos</p> <ul style="list-style-type: none"> • Concepto archivo y registro, relación entre archivo y registro, aplicación de los archivos • Operaciones sobre archivos: crear, abrir, cerrar un archivo. • Agregar, modificar, eliminar y consultar datos de un archivo.
14	12 – 18 Noviembre	Módulo 5 - Continuación
15	19 – 25 Noviembre	Módulo 5 - Continuación
16	26 – 02 Noviembre	Entrega y defensa de II Tarea Programada
17	03 – 09 Diciembre	Ampliación
18	10 – 16 Diciembre	

Bibliografía

- Deitel, P. J., & Deitel, H. M. (2015). *Cómo programar en Java (10 ed.)*. México: Pearson Education.
- Deitel, P. J., & Deitel, H. M. (2017). *Java How to Program, Early Objects (11 ed.)*. US: Pearson Education.
- Deitel, P. J., & Deitel, H. M. (2017). *Java 9 for Programmer (4 ed.)*. US: Pearson Education.
- Schildt, H. (2017). *Java: A Beginner's Guide (7 ed.)*. US: McGraw-Hill Education.
- Schildt, H. (2017). *Java: The Complete Reference (10 ed.)*. US: McGraw-Hill Education.
- Ferreira, W. (2017) *Computer Science Distilled: Learn the Art of Solving Computational Problems*. United States: Code Energy LLC
- 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.



Universidad de Costa Rica
Sede de Occidente
Bachillerato en Informática Empresarial

