



Programación 1

1. Características generales

Nombre:	Programación 1
Sigla:	CI-0112
Créditos:	4
Horas lectivas:	5
Requisitos:	CI-0110 Introducción a la Computación
Correquisitos:	-
Clasificación:	Curso propio
Ciclo de carrera:	II ciclo, 1er año
Docente(s):	José Antonio Ciccio C.
Datos de contacto:	jose.ciccio@ecci.ucr.ac.cr
Grupo:	9
Semestre y año:	II Ciclo 2020
Horario y lugar de clases:	K 7-9:50, V 7-8:50 (http://www.shorturl.at/kDRT4)
Horario y lugar de consulta:	J 17:30-19:30 (http://www.shorturl.at/kDRT4)
Modalidad:	Virtual

2. Descripción del curso

Este curso fomenta en el estudiante habilidades generales para la resolución de problemas de programación, con énfasis en la etapa de implementación de soluciones. Se usa el paradigma de programación orientado a objetos porque permite solucionar problemas de forma más natural.

3. Objetivos

Objetivo general

El objetivo general del curso es que el estudiante sea capaz de resolver problemas mediante la programación de computadoras utilizando el paradigma de programación orientado a objetos.

Objetivos específicos

Durante este curso cada estudiante desarrollará habilidades para:

1. Explicar el modelo de ejecución de los programas.
2. Analizar problemas y diseñar soluciones básicas orientadas a objetos.
3. Implementar soluciones utilizando el paradigma de programación orientado a objetos.
4. Validar informalmente la ejecución de los programas mediante la definición de un conjunto básico de casos de prueba.





5. Aplicar buenas prácticas de programación.
6. Implementar y manipular estructuras de datos.
7. Implementar algoritmos básicos.
8. Utilizar un ambiente de desarrollo integrado.
9. Corregir errores en programas utilizando un depurador.

4. Contenidos

Objetivo específico	Eje temático	Desglose
8, 9	Herramientas de desarrollo	Herramientas de desarrollo: compilador, depurador, ambiente de desarrollo integrado (IDE).
1	Modelo de ejecución de programas	Rastreo de memoria de programa.
2	Resolución de problemas	Proceso de resolución de problemas, análisis, diseño, implementación, prueba, patrones, técnicas de resolución de problemas: divide y vencerás.
2, 3	Diseño e implementación de soluciones	Conceptualización y definición de clases, atributos de clases, instancias de clases, tipos y variables, parte pública, parte privada.
2, 3	Diseño e implementación de soluciones	Conceptualización e implementación de métodos (sin parámetros, con parámetros de valor, con parámetros de referencia), constructores y destructores.
6	Estructuras de datos	Estructuras de datos estáticas: arreglos de una y más dimensiones.
6	Estructuras de datos	Estructuras de datos dinámicas: lista enlazada y árbol binario.





3, 5, 7	Entrada y salida de datos	Flujos de entrada y salida de datos. Validación de datos.
7	Algoritmos	Estructuras básicas de control: secuenciación, bifurcación, iteración.
7	Algoritmos	Algoritmos básicos de búsqueda como: secuencial y binaria, y algoritmos de ordenamiento como: burbuja, selección e inserción.
7	Algoritmos	Funciones recursivas tales como: factorial, Fibonacci, multiplicación de enteros, potencia de dos números, máximo común divisor de dos números, Torres de Hanoi; además, recorrido, inserción y borrado sobre árboles binarios.
4, 5, 8, 9	Calidad de programas	Casos de prueba básicos: casos extremos, intermedios, inválidos. Buenas prácticas de programación: nombres significativos, convenciones de escritura de nombres (clases, variables, métodos, constantes), disposición del texto (tabulación, delimitadores de bloques, espacios y líneas en blanco), documentación interna, orden de las declaraciones (campos, constructores, métodos), modificadores de acceso (privado y público), uso de constructores.

5. Metodología

En cada semana el estudiante dedicará 12 horas virtuales al curso de Programación. El profesor hará disponibles materiales y actividades de estudio asincrónico tales como documentos, videos o prácticas, que el estudiante consultará durante la semana. En el horario oficial del curso se realizarán videoconferencias sincrónicas donde el profesor aclarará dudas grupales e individuales, para dar seguimiento a las actividades virtuales. Las actividades asincrónicas tienen fechas límite y la entrega tardía es evaluada como se indica en las secciones de metodología y lineamientos.

A programar solo se aprende programando. El profesor marcará el ritmo de avance del curso por medio de las tareas programadas. Es muy importante la realización a cabalidad de las tareas programadas, por cuanto el aprendizaje de la programación exige disciplina y maduración de los conceptos y habilidades. Las tareas deben entregarse en las fechas indicadas, para que esto sea posible el estudiante debe empezar a resolver su tarea





en el momento en que el profesor entrega el enunciado. Es buena política por parte del estudiante, codificar y dejar a punto los ejemplos que se analizan en clase. Las tareas programadas deben realizarse en forma individual.

Este es un curso comprensivo, es decir los conceptos que se han aprendido son necesarios para entender los conceptos que se están estudiando, los estudiantes deben saber que las evaluaciones tomarán en cuenta que el estudiante domina toda la materia desde el inicio del curso hasta el momento de la evaluación. Los trabajos y los exámenes son individuales salvo que el profesor diga explícitamente lo contrario. Los estudiantes pueden comentar entre ellos sobre las tareas pero el material entregado debe haber sido hecho por el estudiante que lo entrega. En caso de copia se castigará por igual al que copia y al que da copia. Tampoco se puede entregar código obtenido por medio de Internet.

Los estudiantes deben inscribirse la primera semana en el sitio del curso en Internet y revisar diariamente el mismo, en él se pondrá el material del curso y el estudiante deberá subir sus tareas, al mismo tiempo se llevará el récord de notas del curso de cada estudiante.

El proceso de enseñanza-aprendizaje del curso estará basado en los siguientes tipos de actividades:

- Lecturas, videos y prácticas independientes de consolidación del conocimiento.
- Discusión de conceptos y realización de ejercicios de asimilación en clase.
- Trabajo colaborativo entre compañeros.

Las horas de consulta del profesor serán dedicadas a la revisión de ejercicios y evacuación de dudas específicas de la materia. El asistente del curso brindará apoyo en la revisión y consulta de ejercicios.

6. Evaluación

Todo trabajo evaluable (a excepción de algunas partes de los exámenes y quices) debe ser entregado electrónicamente en el lugar indicado posteriormente. La nota máxima de 10 puntos se distribuye así:

2 exámenes	40%	Examen 1	20%
		Examen 2	20%
Tareas Programadas	30%	Tarea 1	10%
		Tarea 2	10%
		Tarea 3	10%
Tareas cortas y exámenes cortos	30%		

Los exámenes y las pruebas cortas serán prácticos. Evaluarán la capacidad del estudiante para plantear soluciones adecuadas a problemas dados y especificarlas en programas de cómputo. Esto requiere que el estudiante haya adquirido práctica en el planteamiento de soluciones a problemas, dominio del lenguaje de programación y del ambiente de desarrollo.

Las pruebas cortas se realizarán sin previo aviso. Estas pruebas son individuales a menos que se indique lo contrario. Del total de pruebas realizadas en el curso la nota





más baja (incluyendo las no realizadas por el estudiante) no será tomada en cuenta en la evaluación de esta rúbrica.

Para aprobar el curso el estudiante debe tener 6.75 puntos o más. Si la nota final está entre 5.75 y 6.74 tendrá derecho a realizar un examen de ampliación. En este examen el estudiante deberá obtener una nota mínima de 7.0 para aprobar el curso. En caso de que obtenga una nota menor a 5.75, o de presentar el examen de ampliación con una nota inferior a 7.0 reprobará el curso.

7. Cronograma

- Examen 1: 6 de octubre (semana 9)
- Examen 2: 24 de noviembre (semana 16)

8. Políticas

- Se utilizará la plataforma educativa de la UCR (<https://mediacionvirtual.ucr.ac.cr>) como apoyo del curso para los siguientes aspectos (sin que esto vaya en detrimento de las instrucciones emitidas por el profesor durante las horas lectivas): como medio de comunicación entre los participantes del curso, comunicación de eventos, distribución de materiales, entrega y recepción de asignaciones y soporte a las actividades descritas en la metodología. El estudiante deberá registrarse en la plataforma, proporcionar una foto para permitir la interacción con el profesor y revisar frecuentemente los comunicados.
- El código fuente generado a lo largo del semestre se podrá encontrar en el repositorio del curso: <https://github.com/jciccio/programacion2-20> o en el sitio de mediación virtual.
- Salvo indicación del profesor, la resolución de pruebas cortas, tareas programadas y exámenes es individual y es ilegal utilizar o suministrar parcial o totalmente código escrito por otras personas u obtenido de fuentes de información, como por ejemplo de libros o de Internet, sin la autorización expresa del docente. Si el estudiante incumple con esta política se le asignará un cero como nota en la evaluación y se aplicará lo estipulado en el Reglamento de Orden y Disciplina de los Estudiantes de la Universidad de Costa Rica.
(http://www.cu.ucr.ac.cr/uploads/tx_ucruniversitycouncildatabases/normative/orden_y_disciplina.pdf).
- Durante la ejecución de los exámenes es prohibido que el estudiante se comunique por cualquier medio con otra persona que no sea el docente y que transfiera información de cualquier tipo. De comprobarse el incumplimiento de esta norma el estudiante tendrá calificación 0 en el examen y se aplicará el Reglamento indicado en el punto anterior.





- Durante la ejecución de cualquier evaluación, salvo con instrucción expresa del profesor, es prohibido abandonar el sitio en el que se está realizando el examen o acceder a cualquier otro sitio web.
- Es responsabilidad del estudiante realizar respaldos durante la elaboración de las tareas y de los exámenes en diferentes soportes físicos, para solventar eventuales fallas del equipo.
- Es responsabilidad del estudiante mantener en todo momento una versión de la aplicación que está programando (tarea o examen) que compile sin errores. El resultado entregado debe cumplir con esta característica, de lo contrario será calificado con nota 0.
- Adicionalmente a las políticas citadas, la Universidad de Costa Rica norma los procedimientos de evaluación según el Reglamento de Régimen Académico Estudiantil (http://www.cu.ucr.ac.cr/uploads/tx_ucruniversitycouncildatabases/normative/regimen_academico_estudiantil.pdf), particularmente en los artículos 14, 17, 22 y 24.
- La materia a evaluar en los quices y exámenes es acumulativa desde el primer día de clase.
- Las tareas (cortas y programadas) deberán ser entregadas el día y hora propuestos en el enunciado de cada tarea. Cualquier atraso en la entrega de la misma será sancionado de acuerdo a reglas posteriormente descritas.
- Las tareas programadas deberán ser entregadas a más tardar el día y hora indicado por el profesor. Cualquier atraso en la entrega se sancionará de la siguiente forma:
 - Por cada día de atraso, se calificará con base en 10% menos de nota. Por ejemplo, si se entrega con tres días de atraso, se calificará con base en 70.
- Sólo se repondrán evaluaciones en los casos contemplados en el artículo 24 del Reglamento del Régimen Académico Estudiantil de la UCR, según el procedimiento allí descrito.

9. Bibliografía

- Barnes, David y Kolling, Michael. Programación orientada a objetos con Java usando BlueJ. Quinta edición. Pearson. 2013.
- Deitel y Deitel. Java How To Program: Early Objects. Décima edición. Pearson. 2014.
- Wu, C. Thomas. An Introduction to Object-Oriented Programming with Java. McGraw-Hill. Quinta edición. 2009.





- Ceballos, Francisco Javier. Java2: Curso de Programación. Cuarta edición. Editorial Alfaomega. 2011.

10. Aspectos relacionados con el sistema de bibliotecas, acceso a los reglamentos estudiantiles, y evaluación por parte de los profesores

El Sistema de Bibliotecas, Documentación e Información (SIBDI) de la Universidad de Costa Rica (<http://sibdi.ucr.ac.cr/>) cuenta con una amplia gama de recursos de información bibliográfica en diferentes formatos como libros, folletos, publicaciones periódicas, trabajos finales de graduación, entre otros. Algunos de estos recursos se encuentran en Biblioteca Virtual, desde la cual se pueden acceder las publicaciones de conferencias y revistas de ACM, IEEE (<http://sibdi.ucr.ac.cr/dbingenieria.jsp>), o Springer, entre otras. La Biblioteca Luis Demetrio Tinoco ofrece cursos de capacitación a los estudiantes del área de las Ingenierías y Computación.

El sitio web del Consejo Universitario de la UCR contiene las diferentes normativas estudiantiles, que pueden ser consultadas desde el enlace <http://www.cu.ucr.ac.cr/normativa/estudiantil.html>.

Los procedimientos de evaluación y orientación establecidos en el Reglamento de Régimen Académico Estudiantil se encuentran en http://www.cu.ucr.ac.cr/uploads/tx_ucruniversitycouncildatabases/normative/regimen_academico_estudiantil.pdf

Destacamos especialmente los siguientes artículos de dicho Reglamento:

- El Artículo 14 se refiere al contenido que debe tener un programa del curso, incluyendo “las normas de evaluación desglosadas y con las ponderaciones de cada aspecto a evaluar”.
- El Artículo 17 indica que “las normas de evaluación conocidas por los estudiantes pueden ser variadas por el profesor con el consentimiento de la mayoría absoluta (más del 50% de los votos) de los estudiantes matriculados en el curso y grupo respectivo”.
- El Artículo 22 indica que “el profesor debe entregar a los alumnos las evaluaciones calificadas y todo documento o material sujeto a evaluación, a más tardar diez días hábiles después de haberse efectuado las evaluaciones y haber recibido los documentos”.
- El Artículo 24 establece el procedimiento y fechas para realizar la reposición de evaluaciones.





UNIVERSIDAD DE
COSTA RICA

ECCI
Escuela de
Ciencias de la
Computación e
Informática

