

Universidad de Costa Rica
Escuela de Ciencias de la
Computación e Informática
4 créditos, 4 horas presenciales,
8 extra clase
Curso de Servicio

CI-0202
Principios de Informática

Grupo 01: M 18:00 – 21:50
Prof.: MTI. Henry Lizano Mora
Correo: Henry.lizano@ucr.ac.cr
Consultas: L 17 – 19:50

Descripción del curso

Es un curso básico de programación para estudiantes en el área de ingeniería y afines. En el curso se introduce al estudiante al pensamiento abstracto para la resolución de problemas de ingeniería y científicos, automatizable por medio de herramientas informáticas de desarrollo, utilizando las metodologías sistemáticas. El estudiante aprenderá a reconocer la aplicabilidad de flujos de control y modelos de datos básicos para lograr el diseño e implementación de programas y algoritmos.

Objetivos

Proveer formación básica en programación y construcción de algoritmos y de programas, para la resolución de problemas utilizando técnicas actuales. Al finalizar este curso el estudiante será capaz de:

1. Diseñar, organizar e implementar algoritmos para resolver problemas específicos del área de ingeniería, ciencias y afines.
2. Usar un ambiente de programación para la edición, prueba y depuración de programas.
3. Reutilizar componentes de software.
4. Aplicar buenas prácticas de construcción de software.

Contenidos y cronograma

| | |
|--|-------------------------|
| 1. Fundamentos de la programación | 7 – 11 marzo |
| Lenguajes de programación: concepto de programación, lenguaje máquina, lenguaje ensamblador, lenguaje de alto nivel, máquina virtual, compilador y paradigmas. | |
| 2. Introducción a la programación orientada a objetos | 4 – 18 marzo |
| Paradigma: clases e instancias, atributos y métodos, abstracción y reutilización. Análisis y diseño: modelaje de clases e instancias. Compilación y ejecución | |
| SEMANA SANTA | 21 – 25 marzo |
| 3. Sistemas numéricos y representación de datos | 28 mar – 1 abril |
| Bases y conversiones: decimal, binaria y hexadecimal Sistemas de codificación: ASCII y UNICODE | |
| 4. Tipos de datos | 4 – 8 abril |
| Tipos de datos: primitivos (enteros, reales, booleano y caracteres) e hileras Precisión: entero (byte, short, int y long), real (float y double), booleano, carácter e hilera (ejemplo: secuencias de escape) | |
| 5. Definición y utilización de datos | 11 – 16 abril |
| Definición de variables: Declaración: tipo, identificador y dirección | |

| | |
|--|----------------------------|
| <p>Inicialización: tipo primitivo (valor), instancia (referencia) y estado de memoria Asignación y conversión (i.e. type casting)</p> <p>Utilización de variables: Atributos de clase: declaración, ámbito de vida y ocultamiento (encapsulado) Variables locales: declaración y ámbito de vida Estáticas y constantes: declaración u ámbito de vida</p> | |
| 6. Entrada y salida básica, manejo de excepciones | 18 – 22 abril |
| <p>Entrada y salida básica: Entrada: parámetros de línea de comandos y diálogo Salida: línea de comandos y diálogo</p> <p>Manejo de excepciones: Concepto, ejemplos y definición (clase), lanzamiento y atrape</p> | |
| 7. Expresiones y operadores | 25 – 29 abril |
| <p>Aritméticos binarios (multiplicativos y aditivos) y unarios (negación y posfijos) Relacionales (comparación e igualdad), lógicos (binarios y unarios) y asignación Evaluación y orden de precedencia</p> | |
| 8. Instrucciones y estructuras de control | 2 – 6 mayo |
| <p>Estructuras de secuenciación ({} Estructuras de selección o bifurcación (if / else y switch) Estructuras de repetición o iteración (while, do y for)</p> | |
| 9. Métodos: Fundamentos | 9 – 13 mayo |
| <p>Conceptos: modularización y reutilización, declaración e innovación Componentes: encabezado (identificador, parámetros y tipo de retorno) y cuerpo Métodos estáticos (funciones) Sobrecarga: declaración, firmas y resolución de llamados</p> | I examen (hasta tema 8) |
| 10. Métodos: Funcionamiento | 16 – 20 mayo |
| <p>Paso de argumentos: por valor y por referencia Estado de memoria estática, memoria dinámica y pila de llamados Reglas de alcance o ámbito de identificadores Constructores: concepto y utilización, declaración e invocación</p> | |
| 11. Recursividad | 23 – 27 mayo |
| <p>Concepto y utilización Orden de llamados</p> | |
| 12. Arreglos y vectores: fundamentos | 30 mayo – 3 junio |
| <p>Concepto, estructura y estado de memoria Declaración e inicialización Acceso a celdas y recorrido Parámetros de tipo arreglo y paso de argumentos</p> | |
| 13. Arreglos y vectores: operaciones | 6 – 10 junio |
| Utilidad y operaciones comunes (suma, promedio, mínimo, máximo) | II examen |

| | |
|---|-------------------------------|
| Búsqueda: primitivos e instancias Ordenamiento: primitivos e instancias | (hasta tema 10) |
| 14. Matrices | 13 – 17 junio |
| Concepto, estructura y estado de memoria Declaración e inicialización Acceso a celdas y recorrido | |
| 15. Hileras o cadenas de caracteres (textos) | 20 – 24 junio |
| Concepto Operaciones: concatenación, obtener tamaño, extraer carácter o fragmento, comparación, búsqueda, reemplazo, conversión a mayúscula o minúscula, conversión a arreglo. | |
| 16. Entrada y salida: archivos | 27 junio – 1 julio |
| Conceptos y organización física de archivos Operaciones de archivos: lectura y escritura Procesamiento de binario/textual: apertura – cierre y lectura – escritura | |
| 17. Programación avanzada | 4 – 8 julio |
| Desarrollar uno de los siguientes temas: Matrices, algoritmo y bibliotecas de algebra lineal Fundamentos de graficación y de interfaces graficas Punteros y referencias, copia y clonación de objetos Algoritmos de búsqueda y ordenamiento básico Herencia y polimorfismo | III examen (hasta tema 14) |

Metodología y actividades

En cada semana se imparten dos lecciones, una magistral y otra en laboratorio de cómputo. En las lecciones magistrales se utilizan recursos audiovisuales y la pizarra para ilustrar conceptos e implementaciones. En las lecciones de laboratorio los estudiantes resuelven e implementan ejercicios cortos de programación relacionados con la teoría impartida en las clases magistrales.

El estudiante podrá profundizar individualmente los contenidos cubiertos a través de tareas cortas extra clase. La entrega de dichas evaluaciones se realizará a través del aula virtual alojada en mediación virtual (<http://mediacionvirtual.ucr.ac.cr>). El profesor proveerá en las primeras lecciones del curso, indicaciones a los estudiantes para acceder al aula virtual.

La logística para realizar los exámenes parciales, por ejemplo, en papel o herramientas digitales, es escogido por el profesor y comunicado a los estudiantes durante las primeras lecciones del curso. El profesor indicara además las reglas para anunciar los exámenes cortos (quices).

Evaluación

La evaluación se realizará a través de exámenes parciales, exámenes cortos y tareas cortas con la ponderación mostrada en la siguiente tabla. Los exámenes parciales se aplicarán en tiempo de clase a criterio y organización del profesor, la nota N de una tarea entregada h horas tarde se calculará como $N = x \cdot \frac{3}{2}$, donde x es la nota de habría obtenido si se hubiere entregado a tiempo. En cualquier evaluación donde se detecte plagio, su calificación será anulada por completo y se aplicará la reglamentación correspondiente.

| Actividad | Peso |
|----------------------|------|
| 3 exámenes parciales | 60% |
| Exámenes cortos | 20% |
| Tareas cortas | 20% |

Bibliografía

- Barnes, D., & Kolling, M. (2011). *Objects First with Java: A Practical Introduction Using BlueJ* (5ta ed.). Prentice Hall.
- Ceballos, F. (2010). *Java 2. Curso de Programación* (4ta ed.). Ra-Ma.
- Deitel, P., & Deitel, H. (2014). *Java: Cómo Programar*. Prentice Hall.
- DrJava. (3 de 1 de 2015). *Dr Java*. Obtenido de DrJava Org: <http://www.drjava.org/>
- JAMA. (3 de 1 de 2015). *JAMA: A Java Matrix Package*. Obtenido de JAMA: <http://math.nist.gov/javanumerics/jama/>
- Jeliot. (3 de 1 de 2015). *Jeliot Program Visualization Application*. Obtenido de Jeliot: <http://cs.joensuu.fi/jeliot/>
- Joyanes Aguilar, L. (2001). *Java 2 - Manual de Programación*. McGraw-Hill.
- Schildt, H. (2014). *Java: The Complete Reference* (9th ed.). McGraw-Hill Osborne Media.
- Universidad de Costa Rica. (1 de 3 de 2015). *Mediación Virtual*. Obtenido de Plataforma Institucional de aulas virtuales: <http://mediacionvirtual.ucr.ac.cr/>
- David J Eck, "Introduction to Programing Using Java", Seven Edition. Libro disponible en <http://math.hws.edu/javanotes>

Software

1. Java SE: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html?ssSourceSiteId=otnes>
2. Jeliot 3: <https://cs.joensuu.fi/jeliot/downloads.php>
3. BlueJ: <http://www.bluej.org/>
4. Greenfoot: <http://www.greenfoot.org/door>
5. JAMA: <http://math.nist.gov/javanumerics/jama/>
6. Chart2D: <http://chart2d.sourceforge.net/>
7. Alice: <http://www.alice.org/index.php>