

Modalidad virtual. Curso de servicio, sin requisitos ni correquisitos. 4 créditos, 4 horas virtuales asíncronas, 8 horas extra-clase

Consulta: M 17:00-19:00 via Zoom previa cita por Telegram/Mediación Virtual.

Aula virtual: Mediación Virtual.

Mensajería instantánea: Telegram.

Descripción del curso

Es un curso básico de programación para estudiantes del 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 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

Semana

1. Fundamentos de la programación	10-ago
Lenguajes de programación: concepto de programación, lenguaje máquina, lenguaje ensamblador, lenguaje de alto nivel, máquina virtual, compilador y paradigmas. Ciclo de vida de un programa: problema, análisis, diseño, implementación y prueba Algoritmo: concepto, primitivas y ejemplos	
2. Introducción a la programación orientada a objetos	17-ago
Paradigma: clases e instancias, atributos y métodos, abstracción y reutilización Análisis y diseño: modelaje de clases e instancias Construcción (compilación, interpretación) y ejecución	
	(17-ago feriado)
3. Sistemas numéricos y representación de datos	24-ago
Bases y conversión: decimal, binaria y hexadecimal Sistemas de codificación de texto: ASCII y UNICODE	
4. Tipos de datos	24-ago
Tipos de datos: enteros, reales, booleanos, caracteres e hilas de texto. Aritmética de precisión entera, aritmética de precisión flotante.	
5. Definición y utilización de variables	24-ago
Definición de valores y variables: Declaración: tipo, identificador y dirección Inicialización de acuerdo al tipo de datos: valores, indirección (ej: punteros, referencias) Estado de la variable en memoria (valor) Asignación y conversión de tipos	
Utilización de variables: Atributos de clase: declaración, ámbito de vida y ocultamiento (encapsulamiento) Variables locales: declaración y ámbito de vida Estáticas y constantes: declaración y ámbito de vida	

6. Entrada y salida básica, verificación	31-ago
Entrada y salida básica: Entrada: parámetros de línea de comandos y entrada interactiva Salida: salida estándar y salida interactiva Verificación de datos y manejo de errores (ej: manejo de excepciones)	
7. Expresiones y operadores	31-ago
Evaluación de expresiones y orden de precedencia Aridad de operadores: unarios (ej: negación) y binarios (ej: multiplicativos y aditivos) Tipos de operadores: aritméticos, relacionales (comparación e igualdad), lógicos, y de asignación	
8. Instrucciones y estructuras de control	07-set, 14-set
Secuencia de instrucciones y bloques de instrucciones Condición de instrucciones Repetición de instrucciones por condición y por contador	
9.a. Subrutinas: fundamentos	21-set
Conceptos: modularización y reutilización, declaración e invocación Componentes: encabezado (identificador, parámetros y tipo de retorno) y cuerpo Funciones y procedimientos Sobrecarga de subrutinas: declaración y resolución de llamados	
9.b. Subrutinas: funcionamiento	28-set
Paso de argumentos: por valor y por indirección Alojamiento estático de memoria, alojamiento dinámico de memoria, y alojamiento automático de memoria (pila de llamados) Reglas de alcance o ámbito de identificadores Constructores: concepto y utilización, declaración e invocación	
10. Recursividad	05-oct
Concepto y utilización Orden de llamados	
11a. Colecciones lineales de datos: fundamentos	12-oct
Concepto (ej: arreglos, listas), estructura y estado de memoria Declaración e inicialización Acceso a elementos y recorrido Paso de colecciones por argumento a subrutinas	
11b. Colecciones lineales de datos: operaciones	19-oct
Utilidad y operaciones comunes (suma, promedio, mínimo, máximo) Búsqueda (de acuerdo al tipo de datos de los elementos) Ordenamiento (de acuerdo al tipo de datos de los elementos)	
12. Matrices	26-oct
Concepto, estructura y estado de memoria Declaración e inicialización Acceso a elementos y recorrido	
13. Hileras o cadenas de caracteres (textos)	02-nov
Concepto Operaciones: concatenación, obtener tamaño, extraer carácter o fragmento, comparación, búsqueda, reemplazo, conversión a mayúscula o minúscula, conversión a arreglo	
14. Entrada y salida: archivos	09-nov
Conceptos y organización física de archivos Operaciones de archivos: lectura y escritura Procesamiento binario/textual: apertura/cierre y lectura/escritura	
15. Temas adicionales	16-nov, 23-nov
Sistemas numéricos y representación de datos.	

Recursividad.
 Colecciones lineales de datos: operaciones.
 Matrices, algoritmos y bibliotecas de álgebra lineal.
 Fundamentos de graficación y de interfaces gráficas.
 Indirección (punteros y referencias), copia y clonación de objetos.
 Algoritmos de búsqueda y ordenamiento básicos.
 Herencia y polimorfismo
 Aritmética de precisión arbitraria.
 Cuadernos de lenguajes de computación.
 Visualización de datos (ej: creación de gráficas, animaciones).
 Simulación (ej: a partir de modelos matemáticos, físicos).
 Predicción (usando modelos simples).

II examen
 (hasta tema 14)

Metodología y evaluación

Se sigue una metodología constructivista de aula invertida. En cada semana el estudiante dedica 12 horas al curso de Principios de informática, en la que consultará vídeos y otros materiales dispuestos por el docente con el fin de resolver los problemas planteados para dicha semana. Durante las 4 horas asignadas para lecciones del curso, los estudiantes podrán recibir apoyo del docente y el asistente a través de la plataforma Zoom con la facilidad de *breakout rooms*. El enlace a cada lección será distinto y comunicado con anticipación a través de mensajería instantánea o el aula virtual.

Las sesiones sincrónicas y las de consulta serán grabadas, y podrían hacerse disponibles para la consulta posterior de acuerdo al interés de los estudiantes. Los vídeos grabados, enunciados de las evaluaciones, y las entregas de estas serán a través del aula virtual alojada en Mediación Virtual (<http://mediacionvirtual.ucr.ac.cr/>). El estudiante entregará sus soluciones en el aula virtual antes de la fecha límite estipulada en el enunciado de cada tarea. En caso de una entrega tardía se aplicará un castigo calculado con la siguiente fórmula: $10 * d$ puntos, d = cantidad de días de tardía. El profesor proveerá en las primeras lecciones del curso indicaciones para acceder al aula virtual.

Los rubros por evaluar para la nota del curso son los siguientes:

Rubro	Valor
I Examen Parcial	25%
II Examen Parcial	35%
Laboratorios	25%
Tareas y exámenes cortos	15%

Cada semana, el material de estudio asincrónico, así como la entrega de laboratorios siguen la siguiente programación:

Día de la semana	Actividad
Lunes	1:00 p.m. Todos los materiales de la semana se encuentran disponibles en el aula virtual.
Jueves	5:30 p.m. Sesión sincrónica para aclarar dudas grupales e individuales. La asistencia es opcional.
Domingo	11:59 p.m. Tiempo máximo para entregar el trabajo correspondiente al laboratorio.

La entrega de tareas y exámenes cortos se entregarán según sea especificado en la asignación.

Se realizarán dos exámenes parciales de ponderaciones 25% y 35% respectivamente en las semanas indicadas en el cronograma a la derecha de los contenidos. Los medios digitales y lineamientos para realizar exámenes serán comunicados por el docente oportunamente durante las lecciones del curso, y por tanto tendrán una duración máxima de 1 hora con 40 minutos. El profesor podría realizar actividades adicionales posteriores a cada examen, como entrevistas con efecto en la calificación del mismo. En caso de desfase, el profesor ajustará las fechas del cronograma y negociará las fechas de los exámenes con los estudiantes una vez que se hayan cubierto los contenidos del mismo. Por la naturaleza de los contenidos, los exámenes son acumulativos. Cada examen planteará un problema que el estudiante debe resolver mediante la programación de computadoras. El profesor realizará una práctica para el primer examen parcial con el fin de que los estudiantes puedan familiarizarse con este tipo de evaluación. Solo se reponen evaluaciones en los casos contemplados en el artículo 24 del Reglamento de Régimen Académico Estudiantil de la UCR, utilizando el procedimiento allí descrito.

Las tareas y exámenes cortos tienen el mismo valor para el cálculo de este rubro en la nota del curso. Los laboratorios tienen el mismo valor para el cálculo de este rubro en la nota del curso. De acuerdo al artículo 15 del Reglamento de Régimen Académico Estudiantil dichas reglas quedan establecidas después de este aviso.

Lineamientos

1. Durante los exámenes el estudiante podrá consultar material teórico (libros, apuntes) o código que sea producto de su trabajo en el curso o que haya sido proporcionado por el profesor. Los medios para consultar estos materiales durante el examen serán indicados previamente por el profesor.
2. Es ilegal presentar como propio, código parcial o total escrito por otras personas u obtenido de fuentes de información, como por ejemplo de libros o de Internet, sin la autorización expresa del docente. Los exámenes son estrictamente individuales, y es ilegal violar esta regla a través de aplicaciones de correo electrónico o mensajería. En cualquier asignación en que se sospeche de plagio se aplicará el debido proceso 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).
3. Todo código entregado por el estudiante debe compilar sin errores. De lo contrario será calificado con 0.
4. En cualquiera de los tipos de evaluaciones, el profesor podría proponer actividades opcionales por crédito extra en la nota del curso.

Bibliografía

1. Guttag, John V. "Introduction to Computation and Programming Using Python: With Application to Understanding Data", 2nd edition, The MIT Press, 2016.
<https://mitpress.mit.edu/books/introduction-computation-and-programming-using-python-second-edition>
2. Langtangen, Hans Petter. "A Primer on Scientific Programming with Python", 5th edition, Springer Berlin Heidelberg, 2016.
<https://link.springer.com/book/10.1007%2F978-3-662-49887-3>
3. Rossant, Cyrille, "Learning IPython for Interactive Computing and Data Visualization", ISBN: 978-1782169949, PACKT Books, 2013.
<https://www.packtpub.com/big-data-and-business-intelligence/learning-ipython-interactive-computing-and-data-visualization>
4. Wachenchauser, Rosita, Manterola, Margarita y otros, "Algoritmos y Programación con lenguaje Python", OpenLibra, 2011.
<https://openlibra.com/es/book/algoritmos-y-programacion-con-lenguaje-python>

Software

1. **Repl.it** es un ambiente de desarrollo y repositorio en línea: <https://repl.it/>
2. **Python 3** Descargas: <https://www.python.org/downloads/>
3. Cuaderno **Jupyter** es una aplicación web que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto explicativo. Sus usos incluyen: limpieza y transformación de datos, simulaciones numéricas y mucho más: <http://jupyter.org/>
4. **Anaconda** (incluye Python y Jupyter): <https://www.continuum.io/downloads>