



Planificaciones

7540 - Algoritmos y Programación I

Docente responsable: ESSAYA DIEGO NICOLAS

OBJETIVOS

- Conocer un lenguaje de programación imperativo.
- Aplicar las capacidades que ofrece el lenguaje de programación elegido en el curso, en particular, estructuras de control, estructuras sencillas de datos e implementación de algoritmos elementales.
- Lograr habilidades para desarrollar programas sencillos.
- Aprender un conjunto de algoritmos básicos. Conocer y comprender algoritmos fundamentales utilizados con frecuencia, su desarrollo, análisis de eficiencia y corrección.
- Poder escribir programas de computadora en un lenguaje de programación imperativo. Poder compilar problemas (sencillos) y soluciones.
- Iniciarse en el diseño e implementación de estructuras de datos fundamentales.
- Desarrollar capacidades cognitivas y para la construcción de modelos y abstracción,
- Desarrollar una manera científica de abordar problemas (procedimiento metódico e inclusión del estado del arte),
- Desarrollar capacidad para documentar soluciones y su evaluación crítica, así como una presentación convincente.

CONTENIDOS MÍNIMOS

-

PROGRAMA SINTÉTICO

Concepto de algoritmo.

Concepto de programa.

Lenguajes de programación.

Tipos de datos simples.

Estructuras de control.

Modularización.

Tipos de datos estructurados.

Algoritmos básicos de ordenamiento y búsqueda.

Introducción a la complejidad computacional.

Archivos.

Tipos de datos abstractos.

Recursividad.

PROGRAMA ANALÍTICO

UNIDAD 1

Concepto de algoritmo. Introducción al manejo de la abstracción: del problema al algoritmo. Concepto de computadora y nociones básicas de su organización interna. Concepto de programa. Concepto de lenguaje de programación. Nociones y diferencias básicas entre compiladores e intérpretes. Estado y computación. Definición y uso de los conceptos de: tipo de dato, expresión, declaración, definición, sentencia. Entrada/ salida.

UNIDAD 2

Tipos de datos simples: enteros, reales, caracteres, lógicos; primitivas de cada tipo. Concepto y uso de constantes y variables. Operadores aritméticos. Prioridad de operaciones. Tipo de dato puntero. Definición de tipos.

UNIDAD 3

Sentencias de asignación, de bifurcación, de elección múltiple, de control de ciclos. Diseño y verificación de algoritmos. Introducción a las prácticas de documentación, verificación y prueba de programas. Manejo de errores. Introducción a los archivos de texto como entrada / salida de programas sencillos.

UNIDAD 4

Introducción a la modularización de programas mediante funciones. Conceptos de cohesión y acoplamiento de subprogramas. Ámbito de las variables y análisis de las prácticas correctas en los pasajes de las mismas a los módulos de un programa. Parámetros formales y reales. Pre y postcondiciones. Introducción a los métodos de división y conquista.

UNIDAD 5

Tipos de datos estructurados provistos por el lenguaje: arreglos unidimensionales y multidimensionales, cadenas, registros.

UNIDAD 6

Algoritmos básicos de ordenamiento y búsqueda en arreglos: selección, inserción, burbujeo, búsqueda secuencial, búsqueda binaria. Introducción a la complejidad computacional. Nociones de tiempo y espacio consumido. Comparación de algoritmos en cuanto a la complejidad computacional.

UNIDAD 7

Archivos: conceptos básicos, operaciones con archivos de texto y binarios. Procesamiento de archivos secuenciales y de acceso directo. Corte de control. Apareo de archivos. Manejo de índices.

UNIDAD 8

Iniciación al concepto de tipo de dato abstracto. Tipos de datos abstractos lista, pila, cola. Usos. Nociones básicas de recursividad.

BIBLIOGRAFÍA

Cátedra de Algoritmos y Programación I. Apuntes de cátedra.

John Zelle: \Python Programming: An Introduction to Computer Science, Franklin, Beedle & Associates, 2003.

Allen B. Downey: \Think Python: An Introduction to Software Design",2008.
<http://www.greenteapress.com/thinkpython/>

Mark Lutz: \Learning Python (3rd Edition), OReilly Media Inc.2007.
An Introduction to Python Paperback – March 1, 2011
by Guido Van Rossum , Fred L Drake Jr

N. Wirth: Algorithms + Data Structures = Programs. Editorial Perentice Hall.

N. Wirth: Introducción a la Programación Sistemática. Editorial El Ateneo.

B. Kernighan, D. Ritchie: El lenguaje de programación C. Editorial Pearson Educación

Apunte del curso: <https://algoritmos1rw.ddns.net/material>

RÉGIMEN DE CURSADA**Metodología de enseñanza****Clases teórico-prácticas**

Exposición teórica de conceptos fundamentales, con estudio de casos y resolución metódica de problemas tipo con ensayos sobre objetivos.

Clases prácticas

Resolución por parte de los alumnos -y orientada por los docentes- de problemas correspondientes a las unidades temáticas del programa, ya sea por escrito o por computadora. En general se tratará de problemas abiertos, que generen dudas y motiven la consulta a los docentes y la profundización del conocimiento a través de la bibliografía. Durante el curso se plantearán trabajos prácticos con problemas a resolver por programación, que los alumnos deberán desarrollar en grupo.

Modalidad de Evaluación Parcial

Desarrollo de trabajos prácticos por computadora.

El examen parcial es de tipo teórico-práctico e incluye la resolución de ejercicios, con dos oportunidades de recuperación.

La evaluación integradora escrita, de carácter teórico-práctico, con dos oportunidades de recuperación.

CALENDARIO DE CLASES

Semana	Temas de teoría	Resolución de problemas	Laboratorio	Otro tipo	Fecha entrega Informe TP	Bibliografía básica
<1> 09/03 al 14/03	Introducción a la materia. Concepto de algoritmo. Primeros pasos en Python. Sintaxis y gramática de Python. Variables. Funciones.					
<2> 16/03 al 21/03	Construcción de programas: análisis, especificación y diseño. Calidad de software. Expresiones booleanas. Operadores lógicos. Estructuras condicionales.					
<3> 23/03 al 28/03	Ciclos. Ciclos indefinidos. Uso de centinelas. Secuencias: Cadenas, tuplas y listas.					
<4> 30/03 al 04/04	Secuencias. Listas y mutabilidad. Buscar en listas, algoritmos de búsqueda (lineal, binaria). Pasaje de funciones como parámetro.					
<5> 06/04 al 11/04	Map y filter. Listas por Comprension. Contratos: pre y post condiciones. Invariantes.					
<6> 13/04 al 18/04	Diccionarios. Parcial.				X	
<7> 20/04 al 25/04	Manejo de archivos. Excepciones. Manejo de errores.					
<8> 27/04 al 02/05	Recorridos de archivos: apareo y corte de control. Objetos. Propiedades. Creación de objetos. Abstracción.					
<9> 04/05 al 09/05	Listas enlazadas.					
<10> 11/05 al 16/05	Pilas y colas. Recursión.					
<11> 18/05 al 23/05	Métodos de ordenamiento sencillos.					
<12> 25/05 al 30/05	División y conquista. Ordenamientos recursivos.					
<13> 01/06 al 06/06	Lenguaje de programación C.					
<14> 08/06 al 13/06	Lenguaje de programación C.					
<15> 15/06 al 20/06	Recuperatorios.					
<16> 22/06 al 27/06	Recuperatorios.					

CALENDARIO DE EVALUACIONES

Evaluación Parcial

Oportunidad	Semana	Fecha	Hora	Aula
1º	6	13/04	17:30	
2º	16	22/06	19:00	
3º		04/12	19:00	
4º				