

GOBIERNO CONSTITUCIONAL DEL ESTADO LIBRE Y SOBERANO DE OAXACA
INSTITUTO ESTATAL DE EDUCACIÓN PÚBLICA DE OAXACA
COORDINACIÓN GENERAL DE PLANEACIÓN EDUCATIVA
COORDINACIÓN GENERAL DE EDUCACIÓN MEDIA SUPERIOR Y SUPERIOR

PROGRAMA DE ESTUDIOS

NOMBRE DE LA ASIGNATURA	DISEÑO ESTRUCTURADO DE ALGORITMOS
-------------------------	-----------------------------------

CICLO PRIMER SEMESTRE	CLAVE DE LA ASIGNATURA PI-01	TOTAL DE HORAS 80
--------------------------	---------------------------------	----------------------

OBJETIVO(S) GENERAL(ES) DE LA ASIGNATURA
Ser capaz de plantear metodológicamente la solución de problemas a través del manejo de técnicas de diseño estructurado y formulación de algoritmos.

TEMAS Y SUBTEMAS
1. CONCEPTOS BÁSICOS 1.1. Definición de lenguaje de programación 1.2. Lenguajes de programación 1.3. Algoritmos 1.4. Definición de algoritmo 1.5. Representación de algoritmos 1.6. Algoritmos cotidianos 2. METODOLOGÍA PARA LA SOLUCIÓN DE PROBLEMAS 2.1. Definición del problema 2.2. Análisis de los datos 2.3. Diseño de la solución 2.4. Codificación 2.5. Prueba y depuración 2.6. Documentación 2.7. Mantenimiento 3. TÉCNICAS DE DISEÑO 3.1. Top down 3.2. Bottom up 4. ENTIDADES PRIMITIVAS 4.1. Tipos de datos

- 4.1.1. Simples
- 4.1.2. Estructurados
- 4.2. Identificadores
- 4.3. Variables
- 4.4. Constantes
- 4.5. Operadores y operandos
- 4.6. Expresiones

5. TÉCNICAS PARA LA FORMULACIÓN DE ALGORITMOS

- 5.1. Pseudocódigo
 - 5.1.1. Palabras reservadas
 - 5.1.2. Operaciones de asignación
 - 5.1.3. Contadores
 - 5.1.4. Acumuladores
 - 5.1.5. Interruptores
- 5.2. Diagramas de flujo
 - 5.2.1. Símbolos utilizados
 - 5.2.2. Estructura de un diagrama de flujo

6. ESTRUCTURAS ALGORÍTMICAS

- 6.1. Secuenciales
 - 6.1.1. Asignación
 - 6.1.2. Entrada
 - 6.1.3. Salida
- 6.2. Condicionales
 - 6.2.1. Simple
 - 6.2.2. Doble
 - 6.2.3. Anidada
 - 6.2.4. Múltiple
- 6.3. Repetitivas o iterativas
 - 6.3.1. Hacer - mientras
 - 6.3.2. Repetir - hasta
 - 6.3.3. Hacer - para

7. ESTRUCTURA DE DATOS

- 7.1. Arreglos
 - 7.1.1. Arreglo unidimensional
 - 7.1.2. Arreglo bidimensional
 - 7.1.3. Arreglo multidimensional
 - 7.1.4. Arreglo de caracteres
 - 7.1.5. Aplicaciones con arreglos
- 7.2. Estructuras

ACTIVIDADES DE APRENDIZAJE

En la unidad 1, el alumno deberá comprender la historia, significado y concepto de algoritmo, así como desarrollar y aplicar el concepto de algoritmo en actividades de la vida diaria. Realizar una práctica breve de programación para algún algoritmo específico, seleccionado ya sea por el alumno o sugerido por el profesor.

En la unidad 2, el alumno debe visualizar algún problema cotidiano y propondrá la solución (su diseño) para éste.

En la unidad 3, facilitar la comprensión del programa, y además permiten, de forma rápida, las ampliaciones y modificaciones que surjan en la fase de explotación del ciclo de vida de un programa o una ampliación informática.

En la unidad 4, realizar operaciones de intercambio de valores, fórmulas y expresiones matemáticas a formas aceptadas por la computadora, además de programar expresiones matemáticas con diferentes tipos de datos y operadores.

En la unidad 5, explicar los métodos de representación y formulación de algoritmos, así como sus ventajas y desventajas: programar algoritmos de casos específicos usando la secuencia de diagramas de flujo N-S y pseudocódigos.

En la unidad 6, explicar las estructuras algorítmicas de secuencia y condición, en forma simple o aislada y como parte de problemas más complejos, se recomienda programar, usando las estructuras de secuencia y condicionalidad.

En la unidad 7, explicar los conceptos básicos y tipos de arreglos que existen para diferentes tipos de datos, así como el enfoque usando para su manejo computacional, elaborar algoritmos y programas para la solución de problemas que involucren arreglos de una o más dimensiones; elaborar programas que manejen principalmente datos STRING.

CRITERIOS Y PROCEDIMIENTOS DE EVALUACIÓN Y ACREDITACIÓN

Se realizan tres evaluaciones parciales y una evaluación ordinaria final de la asignatura.

Para las evaluaciones parciales, se deberá realizar un examen escrito y se podrá complementar la evaluación con exámenes prácticos, avances de proyectos, tareas, investigaciones y otras actividades académicas previamente aprobadas de acuerdo con la normatividad Universitaria. Queda a criterio del profesor la ponderación de todas las actividades.

Para la evaluación ordinaria final, se deberá realizar un examen escrito y se podrá complementar la evaluación con proyectos, exposiciones, tareas e investigaciones realizadas a lo largo del semestre. Queda a criterio del profesor la ponderación de todas las actividades.

Para la calificación final de la asignatura, se establece la ponderación de las evaluaciones parciales y ordinaria final con base en la normatividad de la Universidad.

BIBLIOGRAFÍA (TIPO, TÍTULO, AUTOR, EDITORIAL Y AÑO)

Básica:

- Estructura de datos y algoritmos. Aho, Alfred V.; Hopcroft, John E.; Ullman, Jeffrey D. Addison Wesley. 1998, 1ª Edición.
- Fundamentos de programación, algoritmos y estructura de datos. Joyanes Aguilar, Luis. McGraw-Hill. 2003, 3ª Edición.
- Introducción al diseño y análisis de algoritmos: un enfoque estratégico. Lee, R.C.T.; Tseng, S.S.; Chang, R.C.; Tsai, Y.T. McGraw-Hill. 2007, 1ª Edición.
- Metodología de la programación: algoritmos, diagramas de flujo y programas. Cairó Battistutti, Osvaldo Gabriel. Alfaomega. 2005, 3ª Edición.

Consulta:

- Algoritmos y estructuras de datos: aproximación en C. Bowman, Charles F. Oxford University Press. 1999.
- Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci. De Giusti, Armando E. Prentice Hall. 2001, 1ª Edición.
- Cómo programar en C++. Deitel, Harvey M.; Deitel, Paul J. Prentice Hall. 2003, 4ª Edición.
- Computación y programación moderna. Levine, Guillermo. Addison Wesley. 2001.
- Fundamentos de computación. Arechiga, V. Rafael. Limusa. 1991, 1ª Edición.
- Fundamentos de programación: algoritmos, estructura de datos y objetos. Joyanes Aguilar, Luis. McGraw-Hill. 2008, 1ª Edición.
- Fundamentos de programación: libro de problemas. Joyanes Aguilar, Luis. McGraw-Hill. 2003, 2ª Edición.
- Introducción a la ciencia de las computadoras. Scheid, Francis. McGraw-Hill. 1984.
- Introducción a la informática. Prieto Espinosa, Alberto; Lloris Ruiz, Antonio; Torres Cantero, Juan Carlos. McGraw-Hill. 2002. 3ª Edición.

- Introducción a la programación con ejemplos en Visual Basic, C, C++ y Java. Mata Toledo, Ramón A. McGraw-Hill. 2001.
- Introducción a la programación: algoritmos y su implementación en VB.NET, C#, Java y C++. Ramírez, Felipe. Alfaomega. 2007.
- Introduction to algorithms. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. The MIT Press. 2001, 2ª Edición.
- Metodología de la programación: aplicaciones en COBOL. Alcalde Lacharro, Eduardo. McGraw-Hill. 1992, 2ª Edición.
- Pascal y Turbo Pascal: un enfoque práctico. Joyanes Aguilar, Luis; Hermoso López, Ángel; Zahonero Martínez, Ignacio. McGraw-Hill. 1994, 1ª Edición.
- Programación en C: metodología, algoritmos y estructura de datos. Joyanes Aguilar, Luis; Zahonero Martínez, Ignacio. McGraw-Hill. 2005, 2ª Edición.
- Programación estructurada en lenguaje C. López Román, Leobardo. Alfaomega. 2005.
- Técnicas de análisis y programación de sistemas: compendio fascicular. Colegio de bachilleres. Noriega Editores. 2004, 1ª Edición.
- Teoría y problemas de computación aplicada a los negocios. Orilia, Lawrence S. McGraw-Hill. 1985.

PERFIL PROFESIONAL DEL DOCENTE

Licenciatura en Informática, Computación, Ingeniería en Sistemas Computacionales o afines, con grado de Maestría y preferentemente de Doctorado en Informática, Ciencias Computacionales o afines. Con experiencia profesional y docente de un año.