**ASIGNATURA : ESTRUCTURA DE DATOS**

**CÓDIGO : PROG-222**

**CRÉDITOS : TRES (3)**

**PRERREQUISITOS : PROG-211**

**UBICACIÓN : 4to Semestre [ISI – LI - TAP]**

**I. DESCRIPCIÓN DEL CURSO**

*Se prepara el estudiante para resolver problemas específicos utilizando estructuras de datos. Utiliza algoritmos para demostrar de forma gráfica el almacenamiento, manipulación y trasformación de los datos en información.*

**II. PROPÓSITO**

* Introducir al estudiante en el conocimiento de las diferentes estructuras de datos tales como listas enlazadas, pilas, colas, arboles; esto con la finalidad de mostrar y fermentar en el estudiante a elegir la estructura adecuada para una aplicación específica.

**III. COMPETENCIAS**

* Usar una herramienta de apoyo a la programación.
* Definir las técnicas de conformación y administración de pilas y colas.
* Identificar las aplicaciones pilas y colas y mostrar las operaciones que ellas pueden realizar.
* Manejar las listas enlazadas como herramientas para el mantenimiento de pilas y colas.
* Aplicar los diferentes tipos de listas enlazadas existentes.
* Aplicar los arboles binarios más usados en la informática.
* Usar las operaciones y algoritmos de grafos de la informática.

**IV. METODOLOGÍA**

|  |  |  |
| --- | --- | --- |
| **CONCEPTUAL** | **PROCIDEMENTAL** | **ACTITUDINAL** |
| * Conceptualización * Discusión estructurada * Enseñanza en grupo * Estudio independiente * Exposiciones temáticas * Trabajo dirigido | * Aprendizaje basado en el problema * Comprensión de lectura * Demostraciones * Simulaciones | * Actitud Critica * Participación Creadora |

**V. RECURSOS**

* Laboratorio de computo
* Software de aplicación
* Multimedios
* Material fotocopiado
* Material on-line
* Editores en línea
* Compiladores libres

**VI. CONTENIDOS**

**Tema 1.- Introducción a la programación C**

**Objetivos:** Introducir a los estudiantes en la metodología, sintaxis y semántica de la programación en el lenguaje C, así como las herramientas necesarias para codificar, compilar y ejecutar un programa C.

1. Estructura de programas en C
   1. Archivos de encabezado, declaración de constantes, declaración de funciones, función main
2. Tipos de datos y su representación en C
   1. Enteros, flotantes, enteros largos, flotantes largos, caracter, booleanos
   2. int, long, float, double, char, unsigned, void, short
3. Funciones de entrada/salida
   1. printf, scanf, placeholders (parámetros de sustitución)
4. Expresiones, operadores y expresiones
   1. Aritméticas, lógicas, relacionales, asignación, manipulación de bits
   2. Precedencia de operadores
5. Ejemplos de programas simples en C
   1. Hallar el área de un triángulo
   2. Hallar el área y la longitud de la circunferencia de un círculo dado el radio
   3. Hallar el volumen de una esfera
   4. Hallar la distancia entre dos puntos dadas sus coordenadas
6. Instrucciones de selección en C
   1. if (<condición>) [{] <verdad>; [}] [else] [{] <falso>; [}]
   2. switch(<expresión>) { case x: ; case y: default: }
   3. Ejemplos de programas con decisiones en C
   * Calcular la velocidad de un vehículo dadas la distancia y el tiempo

* Hallar las raíces de una ecuación de 2do. Grado
* Determinar si tres valores dados corresponden a los lados de un triángulo rectángulo
* Determinar si un número dado es par o impar
* Determinar si un número dado es múltiplo de otro número dado

1. Instrucciones de repetición en C
   1. while (<condición>) [{] <instrucciones>; [}]
   2. do [{] ... [}] while (<condición>);
   3. for(<inicialización>;<condición>;<cambio a variables>) [{] … [}]
   4. Ejemplos de programas con ciclos en C
      * + Escribir los primeros 20 números enteros positivos
        + Escribir los números pares positivos menores que 40
        + Escribir los números de la serie 3,6,9,12, ..., 99
        + Escribir los primeros 100 números de la serie de Fibonacci (0,1,1,2,3,5,8,...)
        + Leer letras y escribir si es consonante o vocal. Terminar cuando se escriba un dígito
   5. Funciones en C
      * + Parámetros: por valor, por referencia
        + Retorno de valores

**Tema 2.- Bytes, arreglos, punteros y estructuras**

**Objetivos:** Conocer los detalles del manejo de múltiples datos usando arreglos, así como la manipulación de bits, la agrupación de datos de tipos diferentes y el manejo de las direcciones de memoria en las que éstos se localizan.

1. Uso de un byte como una estructura de datos
   1. Manipulación de bits
   2. Inserción de bits
   3. Selección de bits
   4. Corrimientos de bits
2. Arreglos
   1. Declaración de arreglos unidimensionales y multidimensionales
   2. Manipulación de elementos de arreglos
   3. Ejemplos de programas C con arreglos
3. Punteros
   1. Características de una variable: valor y dirección
   2. Definición de puntero
   3. Punteros simples, dobles, triples, etc.
4. Estructuras
   1. Definición
   2. Agrupación de datos de diferentes tipos
   3. Operadores de membresía
   4. Punteros a estructuras
5. Asignación dinámica de memoria

**Tema 3.- Pilas y Colas**

**Objetivos:** Definir las técnicas de conformación y administración de pilas y colas. Identificar las aplicaciones de estas y mostrar las operaciones que con ellas se pueden realizar.

1. Pilas
   1. Definición
   2. Manejo de la pila: UEPS (LIFO)
   3. Implementación de pilas con arreglos
   4. Operaciones con pilas: Push, Pop
2. Colas
   1. Definición
   2. Manejo de la cola: PEPS(FIFO)
   3. Implementación de colas con arreglos
   4. Operaciones con colas: Push, Pop

**Tema 4.- Listas enlazadas**

**Objetivos:** Definir los conceptos de listas enlazadas, manejo de las listas enlazadas como herramientas para el mantenimiento de pilas y colas. Mostrar los diferentes tipos de listas enlazadas existentes y su utilización de acuerdo a la aplicación.

1. Listas enlazadas
   * Definición
   * Nodo: dato, enlace
   * Operaciones con listas: Agregar, borrar, insertar, ordenar
2. Listas simplemente enlazadas
3. Listas doblemente enlazadas
4. Listas circulares simplemente enlazadas
5. Listas circulares doblemente enlazadas
6. Matrices dispersas

**Tema 5.- Recursividad**

**Objetivos:** Definir los conceptos de la recursividad. Mostrar la aplicación de ésta en la programación en lenguaje C. Realizar, usando este recurso, algunos programas tradicionales cuya solución recursiva sea la más apropiada.

1. Recursividad
   1. Definición
2. Ejemplos de funciones recursivas
   1. Hallar el factorial
   2. Hallar el máximo común divisor
   3. Hallar el n-ésimo número de Fibonacci
3. Prueba de funciones recursivas

**Tema 6.- Árboles**

**Objetivos:** Definir la estructura y organización de datos mediante el uso de árboles binarios y árboles Bayer además de los algoritmos necesarios para su implementación.

1. Árboles binarios
   * Definición
   * Operaciones
   * Recorrido: In, pre y post orden
2. Árboles Bayer
3. Ejemplos de programas usando árboles

**Tema 7.- Manipulación de archivos**

**Objetivos.**- Definir la utilización de archivos de datos con el objetivo de lograr la persistencia de datos así como el conocimiento de los algoritmos de organización y búsqueda en dichos archivos.

1. Archivos
   * Definición
   * Persistencia de datos: almacenamiento de datos en memoria secundaria
   * Operaciones: escritura, lectura, búsqueda
   * Recorrido: Secuencial, aleatorio
2. Archivos de bajo nivel: funciones
3. Archivos de alto nivel: funciones
4. Archivos de texto y binarios
5. Búsqueda de datos en archivos: Hashing

**Tema 8.- Grafos**

**Objetivos.-** Conocer las estructuras y algoritmos de manejo de grafos.

1. Grafos
   * Definición
   * Vértice, peso, enlace y ruta
   * Adyacencia de vértices
2. Matriz de adyacencia
3. Listas de adyacencia
4. Recorrido de un grafo
5. Problema del camino más corto entre dos vértices

**VIII. INTEGRACIÓN FILOSÓFICA**

|  |  |
| --- | --- |
| **VALOR** | **INTEGRACIÓN** |
| Honestidad | La honestidad es la base del éxito, cada alumno trabajara en la honestidad de sus tareas, exámenes, proyectos individuales y en equipo |
| Creatividad | Durante el desarrollo de programas, el alumno someterá su creatividad en el desarrollo de estos. |
| Individualidad | En cada uno de sus proyectos que requieran ser originales y aportar sus propios puntos de vista |
| Cooperación | Apoyar a sus compañeros en cada uno de los proyectos integradores |
| Responsabilidad | Entregar a Tiempo las tareas, proyectos, y la puntualidad en la asistencia a clases. |
| Humildad | Ser humilde en cada una de las opiniones de los demás en las discusiones grupales |

**IX. EVALUACIÓN**

|  |  |
| --- | --- |
| Asistencia | 05 |
| Participación | 05 |
| Exámenes | 30 |
| Examen Final | 10 |
| Tareas y asignaciones | 30 |
| Problemario | 14 |
| Reportes de lectura | 06 |
|  |  |
| **TOTAL** | **100** |

**X. BIBLIOGRAFÍA**

**Joyanes, A.L., Zahonerd M.I. (2005).** Programación en C . 2da Ed., Mc Graw Hill. España.

**Bibliografía Sugerente:**

* **Joyanes, A.L., Sánchez, G.L., Fernández A.M., Zahonerd M.I. (2005).** Estructuras de Datos en C, Mc Graw Hill. España.
* **Computadoras y Procesamiento de Datos ,** Julio E. Villanueva-Lara, Editora Eva V. Chesneau