

Actividad 3

Estructuras de Control de Selección

Objetivos

- Evaluar los conocimientos adquiridos sobre expresiones lógicas y estructuras de control de selección.
- Desarrollar habilidades prácticas en la implementación de algoritmos utilizando estructuras condicionales.
- Fomentar el uso de buenas prácticas de programación y optimización de código.
- Practicar la implementación de algoritmos fundamentales mediante problemas reales.

Competencias a Desarrollar

- Programación condicional avanzada.
- Manejo de estructuras de decisión.
- Validación y tratamiento de datos de entrada.
- Resolución de problemas mediante estructuras de control.
- Implementación de menús interactivos.

Requisitos Previos

- Conocimientos básicos de programación.
- Conocimiento de expresiones lógicas y tipos de datos.
- Entorno de desarrollo configurado con GCC o G++.
- Editor de texto configurado (VSCode).

Instrucciones

1. Abre Visual Studio Code y configura tu entorno para compilar programas en C/C++ usando GCC o G++.
2. Completa todos los ejercicios propuestos en lenguaje C/C++.

3. Nombra cada archivo fuente con extensión .cpp según las convenciones establecidas en clase (por ejemplo, iniciales_ACT#_#.cpp, PNY_ACT2_01.cpp).
4. Para cada ejercicio:
 - Captura una imagen del código fuente.
 - Captura una imagen de la ejecución del programa mostrando la salida en consola.
5. Elabora un documento Word que incluya:
 - Una **portada** con el nombre de la actividad, tu nombre, número de control y grupo.
 - Las capturas de pantalla del código.
 - Las capturas de pantalla de las ejecuciones.
 - Comentarios o explicaciones relevantes.
6. Exporta el documento a formato PDF.
7. Nombra el archivo PDF como INICIALES_PE_ACT3.PDF.
8. Entrega en la plataforma CLASSROOM:
 - El archivo PDF.
 - Los archivos fuente (.cpp) de todos los ejercicios.

Ejercicios

1: Menú de Operaciones Básicas

Desarrolle un programa que permita al usuario realizar operaciones matemáticas básicas mediante un menú interactivo. El programa debe solicitar dos números enteros y mostrar un menú con las siguientes opciones:

1. Suma
2. Resta
3. Multiplicación
4. División

El usuario seleccionará una opción del menú, y el programa realizará la operación correspondiente, mostrando el resultado final.

Implementación:

- Utilice la estructura de selección múltiple (switch) para manejar las opciones del menú.
- Valide la entrada del usuario para evitar errores.

2: Conversión de Unidades de Medida

Desarrolle un programa que permita al usuario convertir unidades de longitud mediante un menú interactivo. El programa debe mostrar las siguientes opciones:

1. Centímetros a pulgadas
2. Centímetros a pies
3. Kilómetros a millas
4. Pulgadas a centímetros
5. Pies a centímetros
6. Millas a kilómetros

El usuario seleccionará una opción y proporcionará el valor a convertir. El programa calculará y mostrará el resultado de la conversión.

Implementación:

- Utilice la estructura de selección múltiple (switch) para manejar las conversiones.
- Incluya constantes para los factores de conversión (por ejemplo, 1 pulgada = 2.54 cm).

3: Determinación del Número Mayor

Desarrolle un programa que lea tres números ingresados por el usuario y determine cuál es el mayor de ellos.

Implementación:

- Utilice estructuras condicionales anidadas o múltiples para comparar los valores.
- Asegúrese de manejar correctamente casos donde los números sean iguales.

4: Determinación del Número Medio

Desarrolle un programa que lea tres números ingresados por el usuario y determine cuál es el número medio en valor.

Implementación:

- Compare los números utilizando estructuras condicionales.
- Asegúrese de que el programa funcione correctamente incluso si los números son iguales.

5: Sistema de Evaluación de Calificaciones

Desarrolle un programa que evalúe el rendimiento académico de un estudiante basado en tres calificaciones. El programa debe calcular el promedio y mostrar la evaluación correspondiente según la siguiente escala:

- Promedio < 30: "Repetir".
- $30 \leq$ Promedio < 60: "Extraordinario".
- $60 \leq$ Promedio < 70: "Suficiente".
- $70 \leq$ Promedio < 80: "Regular".
- $80 \leq$ Promedio < 90: "Bien".
- $90 \leq$ Promedio < 98: "Muy Bien".
- $98 \leq$ Promedio ≤ 100 : "Excelente".
- Promedio > 100: "Error en promedio".

Implementación:

- Utilice estructuras condicionales anidadas para evaluar el promedio.
- Usar estructura de árbol, Optimizado.

6: Cálculo del Consumo de Agua

Desarrolle un programa que calcule el costo total del servicio de agua basado en el consumo mensual en metros cúbicos (m^3). El costo se calcula de forma escalonada:

- Rango 1: $0-4 m^3 \rightarrow$ 50 fijo.
- Rango 2: $5-15 m^3 \rightarrow 8.00/m^3$.
- Rango 3: $16-50 m^3 \rightarrow 10.00/m^3$
- Rango 4: $51+m^3 \rightarrow 11.00/m^3$.

El programa debe mostrar el subtotal, el IVA (16%) y el total a pagar.

Implementación:

- Utilice estructuras condicionales para determinar el rango de consumo.
- Calcule el subtotal, el IVA y el total de forma precisa.

7: Promedio Final con Examen Anulado

En la materia de Programación Estructurada, se aplican 5 exámenes. Desarrolle un programa que calcule el promedio final eliminando la calificación menor y considerando únicamente las 4 calificaciones restantes.

Implementación:

- Almacene las calificaciones en un arreglo o lista.
- Identifique la calificación menor y elimínala del cálculo del promedio.
- Muestre el promedio final con dos decimales.

8: Juego de Piedra, Papel o Tijera

Desarrolle un programa que permita a un jugador competir contra la computadora en el clásico juego de Piedra, Papel o Tijera. La computadora seleccionará una opción aleatoria, y el programa determinará el ganador según las reglas del juego.

Implementación:

- Utilice la generación de números aleatorios para la selección de la computadora.
- Maneje diferentes escenarios de juego (empate, victoria del jugador, victoria de la computadora).

Formato del Reporte

El Reporte debe incluir las siguientes secciones:

1. **Portada:** Nombre de la actividad, tu nombre, número de control y grupo.
2. **Introducción:** Breve descripción del propósito de la actividad.
3. **Código:** Incluye bloques de código para cada ejercicio, acompañados de comentarios explicativos.
4. **Resultados:** Capturas de pantalla o texto con los resultados obtenidos al ejecutar los programas.
5. **Conclusiones:** Reflexión sobre lo aprendido y posibles mejoras.

Consideraciones Importantes

- Usa nombres descriptivos para variables y funciones.
- Comenta tu código para facilitar su comprensión.
- Valida las entradas del usuario
- Prueba tu programa con diferentes valores de entrada para asegurar su correcto funcionamiento.
- Mantén una estructura clara y organizada en tu reporte.