**Instituto Superior de Formación Técnica Nº 151 logo151-trans.gif  
Carrera: Analista de Sistemas  
1 Año. Algoritmos y Estructuras de Datos I**

|  |  |
| --- | --- |
| **Trabajo Práctico Nº 3** | **Unidad 3.3** |
| **Modalidad:** Semi-Presencial | **Estratégica Didáctica:** Trabajo Grupal. |
| **Metodología de Desarrollo:** acordar | **Metodología de Corrección:** acordar docente |
| **Carácter de Trabajo:** Obligatorio – con Nota | **Fecha Entrega:** A confirmar por el Docente. |

**Marco Teórico:**

1. ¿Qué entiende por flujo de ejecución?
2. ¿Qué diferencias hay entre Secuencial, Selección y Repetición?
3. Dar ejemplos de las secuencias del punto anterior y su Instrucciones en C++
4. ¿Qué es una condición compuesta? ¿Qué orden de precedencia tienen?
5. ¿Qué es un if “anidado” y que “tipos” de problemas resuelve?
6. ¿Qué es la “Evaluación perezosa?
7. ¿Se pueden anidar bucles? ¿Dar un ejemplo?
8. ¿Qué entiende por visibilidad de los Identificadores? ¿Dar un ejemplo?
9. ¿Qué función cumple la Sentencia Switch?
10. ¿Compararla con la Sentencia IF?
11. La sentencia “Break” produce un salto en la ejecución, podría indicar un caso en el que necesite no incluirla?
12. ¿La sentencia Default se ejecuta en qué momento?
13. ¿Con la data que se amplió de la instrucción Switch, como puede manejar Rangos?
14. ¿Qué es el Shorthand IF? Ejemplifique.

**Marco Práctico:**  
Desarrollar en C++ los siguientes programas:  
(Se recomienda utilizar Funciones, Tipos definidos por el usuario y Enumeradores)

1. Resolver y Realizar en c++   
   1. Hacer un **programa en C++** que convierta centímetros a pulgadas Celsius a Farenheit y libras a kilogramos
   2. Hacer un **programa en C++** que ayude a una empresa a incrementar los salarios de los trabajadores de la siguiente manera:  
        
      Tipo de salario %  
      De 0 a $9 000 20%  
      De $9 000 a $15 000 10%  
      De $15 000 a $20 000 5%  
      Más de $20 000 3%
   3. Una empresa nos pide el armado de una App para la toma de pedidos, primero deberá presentar un menú de Opciones de 4 productos a) Simple b)medio c) calidad d) Premium, luego nos deberá mostrar en menú el tipo de envío: 1) normal 2) express 3) Fast Delivery.  
      A posteriori deberá mostrar por consola el producto seleccionado y el tipo de envío que eligió.  
      PD. Usar Funciones, manteniendo un Main “lo más pequeño posible”, desacoplando las funcionalidades. Trabajar con la visibilidad de los operadores vistos, while, Switch, If ….
   4. Hacer un **programa en C++** que lea dos números x e y, luego hacer la diferencia x - y; si la diferencia es menor o igual a 10, entonces escribir todos los números comprendidos entre y y x

Lic. Oemig José Luis.

**Marco Teórico:**

1. ¿Qué entiende por flujo de ejecución?

El **flujo de ejecución** de un programa es el orden en el que las sentencias se ejecutan.

1. ¿Qué diferencias hay entre Secuencial, Selección y Repetición?

La **ejecución secuencial** es el más básico de los mecanismos de control de flujo y consiste en la ejecución de instrucciones en el orden en que se encuentran en el código fuente del programa. La **selección** consiste en ejecutar instrucciones diferentes dependiendo del resultado de evaluar una expresión o del valor de una variable. La **repetición** es la ejecución reiterada de instrucciones por un número fijo de ocasiones o dependiendo del resultado de evaluar una expresión.

1. Dar ejemplos de las secuencias del punto anterior y sus Instrucciones en C++

**Ejecución Secuencial**:

Las sentencias se ejecutan línea a línea en orden de aparición (de arriba hacia abajo)

Ejemplo:

*double oper1, oper2, prod;*

*cout << "Operando 1: ";*

*cin >> oper1;*

*cout << "Operando 2: ";*

*...*

*cout << "Producto: " << prod;*

**Selección Simple**:

Las estructuras de selección permiten ejecutar diferentes instrucciones dependiendo del valor de una variable o expresión. También se les llama ramificaciones, estructuras de decisión o alternativas.

En C++ se implementa mediante la instrucción *if*

Ejemplo:

*if (condición) {*

*códigoT*

*}*

*[else {*

*códigoF*

*}]*

*condición*: expresión bool

Cláusula *else* opcional

**Selección Múltiple**:

Las estructuras de selección múltiple permiten evaluar expresiones que regresan valores de tipos diferentes al tipo de datos lógico. Cada uno de los posibles valores que puede tomar una variable o expresión se conoce como un caso y para cada caso el programador puede indicar una instrucción diferente.

En C++ esto se implementa mediante la instrucción *switch*

Ejemplo:

switch (num) {

case 4:

{

cout << "Muy alto";

}

break;

case 3:

{

cout << "Alto";

}

break;

case 2:

{

cout << "Medio";

}

break;

case 1:

{

cout << "Bajo";

}

break;

default:

{

cout << "Valor no válido";

}

}

Salida:

Si num == 4 🡪 Muy alto

Si num == 3 🡪 Alto

Si num == 2 🡪 Medio

Si num == 1 🡪 Bajo

**Repetición:**

Las estructuras de repetición, también llamadas bucles o ciclos, permiten ejecutar la misma instrucción cero, una o múltiples veces. Además de la instrucción a ejecutar, tienen una sección de control.

En C++ las estructuras de repetición disponibles son:

Número de iteraciones condicionado (recorrido variable):

* Bucle *while*

*while (condición)*

*{*

*cuerpo*

*}*

Ejecuta el cuerpo mientras la condición sea true

* Bucle *do-while*

*do*

*{*

*cuerpo*

*}while (condición);*

Comprueba la condición al final

Número de iteraciones prefijado (recorrido fijo):

Bucle *for*

*for (inicialización; condición; paso) cuerpo*

Ejecuta el cuerpo mientras la condición sea true

Se usa una variable contadora entera

1. ¿Qué es una condición compuesta? ¿Qué orden de precedencia tienen?

Se denomina **condición compuesta** a toda combinación de condiciones simples y operadores lógicos.

1. ¿Qué es un if “anidado” y que “tipos” de problemas resuelve?

Estructura de Control *if* (anidada)

Las sentencias *if* implementan decisiones que implican una o dos alternativas.

Una sentencia *if* es **anidada** cuando la sentencia de la rama verdadera o la rama falsa es a su vez una sentencia *if*.

Una sentencia *if* anidada se puede utilizar para implementar decisiones con varias alternativas o multi-alternativas

1. ¿Qué es la “Evaluación perezosa?

En la teoría de lenguajes de programación, La evaluación perezosa (del inglés lazy evaluation) o llamada por necesidad es una estrategia de evaluación que retrasa el cálculo de una expresión hasta que su valor sea necesario, y que también evita repetir la evaluación en caso de ser necesaria en posteriores ocasiones.

1. ¿Se pueden anidar bucles? ¿Dar un ejemplo?

Sí. Se pueden anidar bucles. Esto suele utilizarse por ejemplo para recorrer matrices.

Se adjunta ejemplo en el archivo **Matriz.cpp**

1. ¿Qué entiende por visibilidad de los Identificadores? Dar un ejemplo.

La visibilidad de un identificador (variable) depende del ámbito en el que ha sido definida.

El ámbito (en inglés *scope*) es el contexto que pertenece a un nombre dentro de un programa. El ámbito determina en qué partes del programa una entidad puede ser usada.

Esto sirve para que se pueda volver a definir una variable con un mismo nombre en diferentes partes del programa sin que haya conflictos entre ellos.

Si una variable es declarada dentro de un bloque (método/función/procedimiento), ésta será válida sólo dentro de ese bloque y se destruirá al terminar el bloque. Adicionalmente, la variable no podrá verse ni usarse fuera del bloque (en el exterior del bloque). La variable dentro del bloque es una variable local y solo tiene alcance dentro del bloque que se creó y sus bloques hijos, pero no en bloques hermanos ni padres, una variable definida fuera de cualquier bloque es una variable global y cualquier bloque puede acceder a ella y modificarla.

1. ¿Qué función cumple la Sentencia Switch?

La sentencia *switch* sirve para seleccionar entre diferentes opciones para una expresión.

Su formato es:

*switch* (expresión) {

*case* constante1:

{

código1

}

[*break*;]

*case* constante2:

{

código2

}

[*break*;]

...

*case* constanteN:

{

códigoN

}

[*break*;]

[*default*:

{

códigoDefault

}]

}

1. Compararla con la Sentencia IF

La sentencia *Switch* es una estructura de control de selección que permite listar cualquier número de ramas. En otras palabras, es una estructura de control para ramas multivía. Una sentencia *Switch* es similar a sentencias *If* anidadas

La sentencia *switch* evalúa la *expresión*, compara con la lista de los diferentes *case* y, en caso de no ser igual a ninguno de éstos, ejecutará el bloque encabezado por *default*.

La expresión (variable) a evaluar o selector ha de ser un valor con equivalencia **ordinal** (equivalencia numérica entera). Por ejemplo puede ser una variable tipo *int* o *char* (ya que cada *char* lleva un número entero asociado), pero no puede ser un *double* ni un *float* ni una cadena de caracteres (*string*). Sólo se puede evaluar una expresión y no múltiples expresiones.

1. La sentencia “Break” produce un salto en la ejecución, podría indicar un caso en el que necesite no incluirla?

Observando el formato de la sentencia ***switch*** proporcionado en la respuesta del ejercicio 9, si se tienen ***N*** opciones diferentes para ***opciones*** y se decide no colocar ningún ***break***, suponiendo que la constante ***K*** (*K < N*) coincide con ***opciones***, lo que ocurrirá es que se ejecutarán los bloques ***K, K+1, K+2, … , N***, inclusive el bloque correspondiente a ***default***.

Los casos en que quizás no se necesite incluir break puede darse cuando las opciones K+1, K+2, …, N también sean aplicables además de la opción K, es decir cuando las opciones sean “inclusivas” en el sentido de conjuntos ( o “adicionales”).

**Ejemplo**: Supongamos que un videojuego otorga 100 puntos por completar la etapa 1, 200 puntos por completar la etapa 2, 300 puntos por la etapa 3, etc. hasta la etapa 5 (final). Si uno quiere conocer el puntaje acumulado de un jugador, el archivo **switchTest.cpp** tiene la solución.

1. ¿La sentencia Default se ejecuta en qué momento?

La sentencia ***default*** se ejecuta cuando la ***opcion*** no coincidió con ninguno de los ***case*** anteriores.

1. ¿Con la data que se amplió de la instrucción Switch, como puede manejar Rangos?

En general es difícil manejar rangos con la instrucción switch debido a que, como ya se comentó en el punto 10, la expresión que se evalúa en esta instrucción ha de ser ordinal pero tipo *int* o *char*, pero no *double* ni *float*. Por ende, si quiero realizar un programa para saber cuándo aprueba un alumno, siendo que una nota podría ser 5.75, no voy a poder resolverlo fácilmente mediante un switch.

1. ¿Qué es el Shorthand IF? Ejemplifique.

El *shorthand if* es una forma abreviada del *if else* y funciona mediante un **operador ternario** dado que consta de tres operandos.

Sintaxis

variable = (condition) ? expressionTrue : expressionFalse;

Ejemplo:

*int time = 20;*

*if (time < 18) {*

*cout << "Good day.";*

*} else {*

*cout << "Good evening.";*

*}*

Puede abreviarse con el *shorthand if* haciendo:

*int time = 20;*

*string result = (time < 18) ? "Good day." : "Good evening.";*

*cout << result;*