

# Lenguaje de Programación

C++

Estructuras de control: **switch**

José Luis Alonzo Velázquez

Universidad de Guanajuato

Sesión 24

## Estructuras de selección

Aunque la sentencia `if` de C++ es muy potente, en ocasiones su escritura puede resultar tediosa, sobre todo en casos en los que el programa presenta varias elecciones después de chequear una expresión: selección múltiple o multialternativa. En situaciones donde el valor de una expresión determina qué sentencias serán ejecutadas es mejor utilizar una sentencia `switch` en lugar de una `if`.

## Sintaxis de la estructura de control **switch**

```
switch (selector){  
    case <opcion 1>:  
        <bloque de instrucciones>  
        break;  
    case <opcion 2>:  
        <bloque de instrucciones>  
        break;  
        :  
    case <opcion n>:  
        <bloque de instrucciones>  
        break;  
    default:  
        <bloque de instrucciones>  
}
```

Una sentencia **switch** contiene un selector (en el ejemplo, operador), cuyo tipo debe ser int, char o enumerado. Cuando una sentencia **switch** se ejecuta, el valor del selector se compara con las etiquetas case. Si alguna de ellas concuerda con ese valor se ejecutará la correspondiente secuencia de sentencias. Si queremos que varias alternativas tengan el mismo conjunto de sentencias a ejecutar.

## Ejemplo

```
switch (selector){
    case 1:
    case 2:
        printf("Salida para los casos 1 y 2\n");
        break;
    case 3:
        printf("Salida para el caso 3\n");
        break;
    default:
        printf("Salida para los restantes casos\n");
}
```

En este ejemplo, si el selector se evalúa y su valor es 1 ó 2, se ejecuta, en ambos casos, la instrucción printf( "Salida para los casos 1 y 2" ); En este caso particular puede apreciarse la utilidad de **break** a la hora de detener el flujo del programa. La sentencia **switch** puede incluir la opción **default** para establecer la secuencia de sentencias a ejecutar en el caso de que ninguna etiqueta concuerde con el valor de la expresión case. El tipo de esta expresión case y el de las etiquetas tiene que ser el mismo.

## Ejemplo

```
if (operador == +)
    resultado = A + B;
else if (operador == -)
    resultado = A - B;
else if (operador == *)
    resultado = A * B;
else if (operador == /)
    resultado = A / B;
else
    cout << "Operador invalido";
```

## Ejemplo

```
switch (operador) {  
    case + :  
        resultado = A + B;  
        break;  
    case - :  
        resultado = A - B;  
        break;  
    case * :  
        resultado = A * B;  
        break;  
    case / :  
        resultado = A / B;  
        break;  
    default :  
        cout << "Operador invalido";  
}
```

La palabra reservada **break** permite que el flujo de programa se detenga justo después de la ejecución de la sentencia anterior a ese **break**, impidiendo que se ejecuten las sentencias correspondientes a las siguientes alternativas del **switch**. Por tanto, debemos obligatoriamente acabar cada bloque de sentencias correspondiente a cada alternativa con una sentencia **break**.

Por otro lado, la alternativa **default** es opcional y engloba un conjunto de sentencias (que puede ser vacío, contener una sola sentencia o varias) que se ejecutan en caso de que ninguna de las alternativas del switch tenga un valor coincidente con el resultado de evaluar la expresión del selector.

## Problema para clase

Hacer un menu que despliegue lo siguiente en pantalla:

Escoja una opción:

a)opcion 1

b)opcion 2

c)opcion 3

s)salir

si se escoge la opcion 1 imprima "Se eligio la opcion 1",  
analogamente 2 y 3.

-  Como Programar en C/C++, Deitel (Prentice Hall), 2da Edición.
-  Programming Principles and Practice Using C++, Bjarne Stroustrup.
-  <http://www.codeblocks.org>
-  <http://www.wxwidgets.org>
-  (O'Reilly) Practical C Programming (3rd Edition)