Semana-02: Blender
C++: Estructura de decisión múltiple switch-case

switch - case

Es una estructura de **decisión múltiple**, que en ciertos casos favorece el entendimiento y rapidez de la programación.

Partamos de un ejemplo:

Hacer un programa que permita leer por teclado el número de día de una semana (1 al 7), y muestre en pantalla el nombre de dicho día.

Para su solución nos damos cuenta fácilmente que debe usarse una estructura de **decisión múltiple** (*observe la siguiente diapositiva)*

Nota: *Se asume que los programas en lenguaje C++ en este material están escritos en la función* ***main***

Solución 1:

```
int dia;

cout<<"Ingrese el numero de dia de la semana (1-7): ";
cin>>dia;

if(dia==1) {
   cout<<"Estamos Lunes\n";
}
else if(dia==2) {
   cout<<"Estamos Martes\n";
}
else if(dia==3) {
   cout<<"Estamos Miercoles\n";
}
else if(dia==4) {
   cout<<"Estamos Jueves\n";
}
else if(dia==5) {
   cout<<"Estamos Viernes\n";
}
else if(dia==6) {
   cout<<"Estamos Sabado\n";
}
else if(dia==7) {
   cout<<"Estamos Domingo\n";
}
else{
   cout<<"El numero es incorrecto!\n";
}
```

Podemos notar que el dato de la vble. ***dia*** sólo puede tomar valores **enteros DISCRETOS:** 1, 2, 3, 4, 5, 6, ó 7, osea **NO** dentro de un **rango continuo**

Por lo que también se puede usar para la solución, la estructura **switch-case**
(observe la siguiente diapositiva)

Resultado de la ejecución

```
Ingrese el numero de dia de la semana (1-7): 4
Estamos Jueves

--------------------------------
Process exited after 2.783 seconds with return value 0
Press any key to continue . . .
```

Solución 2:

```
int dia;
cout<<"Ingrese el numero de dia de la semana (1-7): ";
cin>>dia;
switch(dia) {
        case 1: cout<<"Estamos Lunes\n";
        break;

        case 2: cout<<"Estamos Martes\n";
        break;

        case 3: cout<<"Estamos Miercoles\n";
        break;

        case 4: cout<<"Estamos Jueves\n";
        break;

        case 5: cout<<"Estamos Viernes\n";
        break;

        case 6: cout<<"Estamos Sabado\n";
        break;

        case 7: cout<<"Estamos Domingo\n";
        break;

        default:
             cout<<"El numero es incorrecto!\n";
        break;
}
```

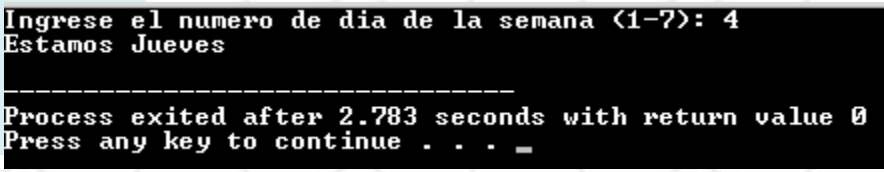
¿Cómo funciona?

En el **caso** que el valor de la vble. ***dia*** sea **1 (case 1:)** , se imprimirá el mensaje *"Estamos Lunes"* y los demás casos se **excluyen** gracias a la instrucción **break;**

En el caso de que el valor de la vble. ***dia*** **NO** sea **1**, se evalúa el siguiente caso **(case 2:)** , y así sucesivamente.

Si ningún caso se cumple, entonces se ejecutará la instrucción por defecto (**default:**), imprimiéndose el mensaje *"El numero es incorrecto!"*

Resultado de la ejecución



Tener cuidado al no poner la instrucción break;

```
int dia;
cout<<"Ingrese el numero de dia de la semana (1-7): ";
cin>>dia;
switch(dia) {
   case 1:  cout<<"Estamos Lunes\n";
   break;

   case 2:  cout<<"Estamos Martes\n";
   break;

   case 3:  cout<<"Estamos Miercoles\n";
   break;

   case 4:  cout<<"Estamos Jueves\n";

   case 5:  cout<<"Estamos Viernes\n";

   case 6:  cout<<"Estamos Sabado\n";

   case 7:  cout<<"Estamos Domingo\n";

   default:
            cout<<"El numero es incorrecto!";
   }
```

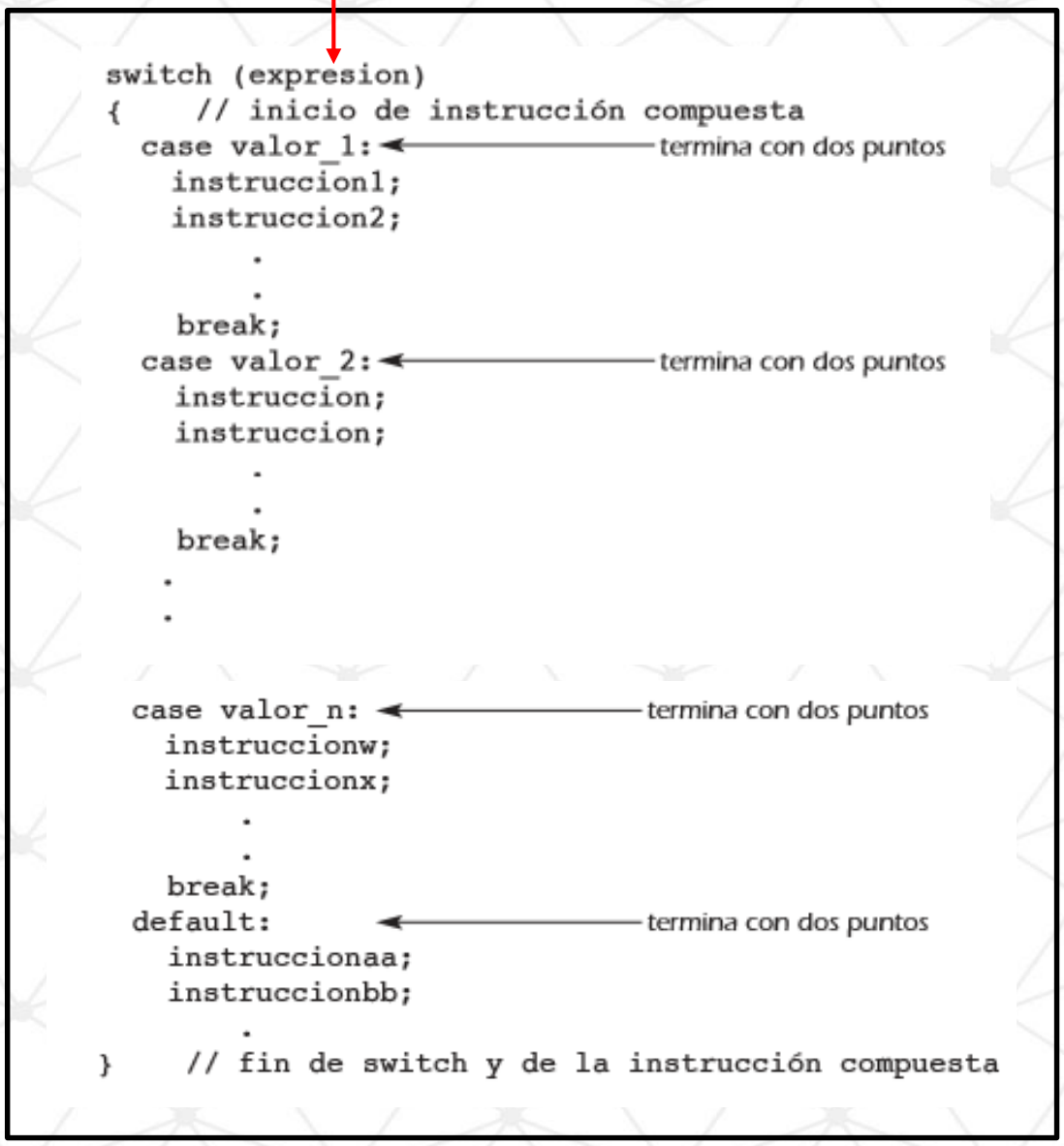
Por ejemplo si el valor de la vble. ***dia*** es **4**, entonces se tendría el siguiente resultado:

Resultado de la ejecución

```
Ingrese el numero de dia de la semana (1-7): 4
Estamos Jueves
Estamos Viernes
Estamos Sabado
Estamos Domingo
El numero es incorrecto!
```

Se cumple el **case 4:** pero al no estar las instrucciones **break;** debajo, los otros casos (**case 5:**, **case 6**:, **case 7**:, incluso **default:**) no se excluyen y son también ejecutados aún no se cumplan.

Importante



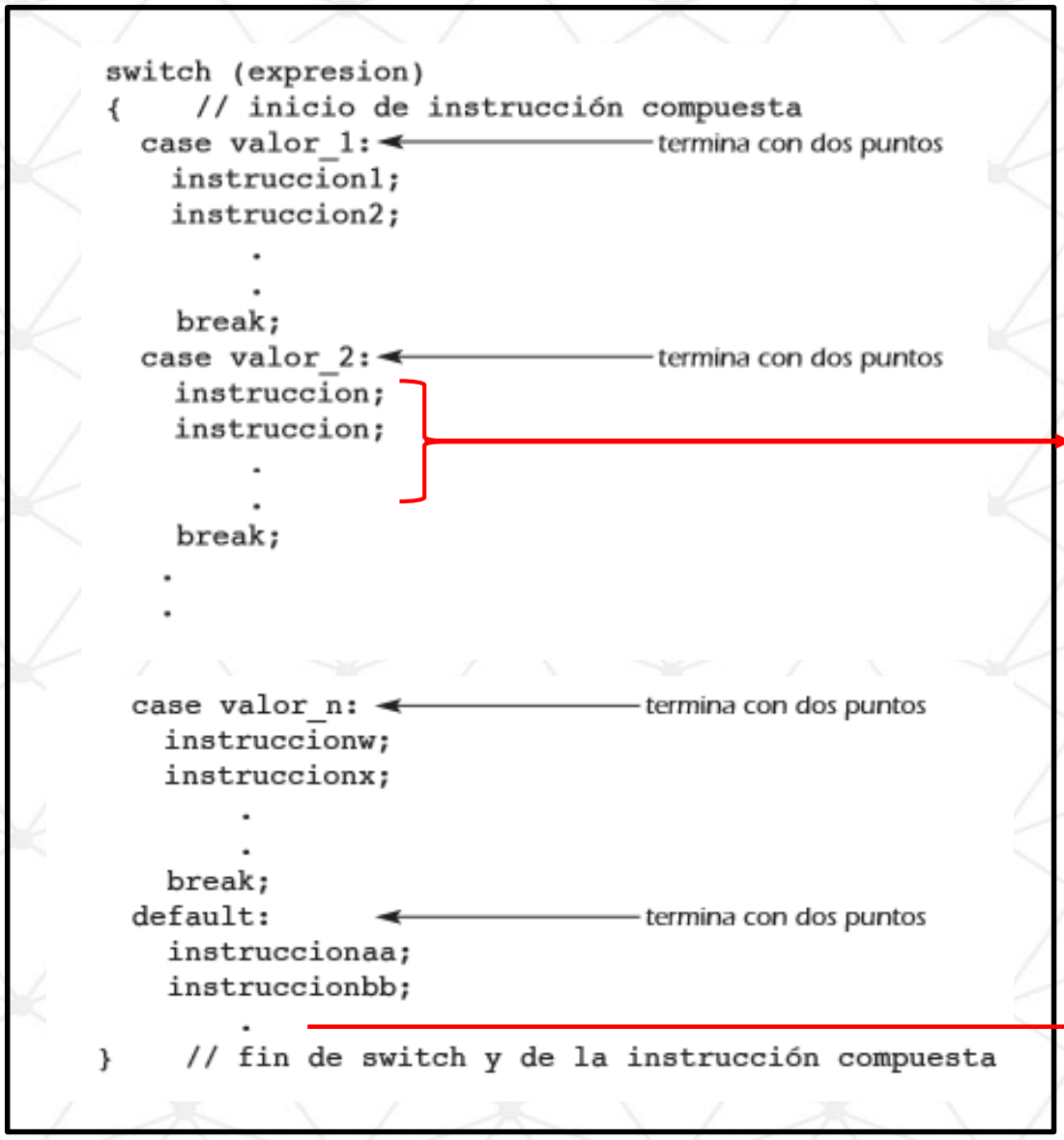
El dato que va en el **() del switch (dato de entrada) tiene que tener** un valor entero, por lo que puede ser también una expresión lógica, una operación entera o un dato tipo **char**

switch-case NO funciona *para datos de entrada* ***float o double***

```
float x = 3.56;

switch(x)
{
   case 3.56: cout<<"mal\n";
   break;

   case 4.2:  cout<<"peor\n";
   break;

   default: cout<<"Horrible\n";
   break;
}
```

INCORRECTO

Importante



Si alguno de los **casos** se cumple, dado el valor de entrada en el **switch**, se puede ejecutar también un **bloque de instrucciones** (no solo una instrucción)

No es obligatorio colocar la instrucción **break;** después de ejecutar las instrucciones por defecto **(default :)**

Importante

switch – case, solo evalúa casos con valor **discreto fijo** (**no desigualdades o rangos**)

Es decir:

```
int x = 3;

switch(x)
{
   case (x>=1):
      cout<<"Bloque 1\n";
   break;

   case (x>10 && x<12):
      cout<<"Bloque 2\n";
   break;

   default:
      cout<<"incorrecto\n";
}
```

INCORRECTO

INCORRECTO

Si se desean evaluar desigualdades o rangos **UTILIZAR estructuras con if…else if….else**

Para finalizar:

Te estarás preguntando: **¿Cúal es el beneficio de usar switch case?**

Pues, ordena mejor el código a la hora de hacer decisiones múltiples, y sobre todo:

Técnicamente su ejecución es mas rápida que las sentencias, if ….else if…..else