Object oriented programming In C++

Control Structure (2)

Professor 최학남

xncui@vision.inha.ac.kr

Office: high-tech 401

Infinite loop

```c
#include <stdio.h>
int main()
{
    int cont = 0,i=1;

    while(1)
    {
        cont = i +cont;
        if(cont>=5050){

        break;
        }

        i++;
    }
    printf("%d,%d\n",i,cont);
}
```

```c
#include <stdio.h>
int main()
{
    int cont = 0,i=1;

    for(;;)
    {
        cont = i +cont;

        if(cont>=5050)
        break;

        i++;
    }
    printf("%d,%d\n",i,cont);
}
```

Compound assignment operators

- +=, -=, *=, /=, %=
  - ✓A+=B ➔ A = A + B
  - ✓A-=B ➔ A = A - B
  - ✓A*=B ➔ A = A * B
  - ✓A/=B ➔ A = A / B
  - ✓A%=B ➔ A = A % B

```c
#include <stdio.h>
int main( ){
    int i = 22 ;
    int a=1,b=3;
    int a1=1,a2=1,a3=1,a4=1,a5=1;
    int c1,c2,c3,c4,c5;
    c1 = a + b;
    c2 = a - b;
    c3 = a * b;
    c4 = a / b;
    c5 = a % b;
    a1 +=b;
    a2 -=b;
    a3 *=b;
    a4 /=b;
    a5 %=b;
    printf("c1 c2 c3 c4 c5\n");
    printf("%2d,%2d,%2d,%2d,%2d\n",c1,c2,c3,c4,c5);

    printf("a1 a2 a3 a4 a5\n");
    printf("%2d,%2d,%2d,%2d,%2d\n",a1,a2,a3,a4,a5);
    return 0;
}
```

```
c1 c2 c3 c4 c5
 4,-2, 3, 0, 1
a1 a2 a3 a4 a5
 4,-2, 3, 0, 1
계속하려면 아무 키나 누르십시오 . . .
```

switch statement

- switch(switch – case – default)
  - ✓The control statement that allows us to make a decision from the number of choices
  - ✓The keyword **case** is followed by an integer or a character constant.

```
switch ( integer expression )
{
        case constant 1 :
                do this ;
        case constant 2 :
                do this ;
        case constant 3 :
                do this ;
        default :
                do this ;
}
```

switch statement

- What happens when we run a program containing a **switch**?
  - ✓ First, the integer expression following the keyword **switch** is evaluated.
  - ✓ The value it gives is then matched, one by one, against the **constant** values that follow the **case** statements.
  - ✓ When a match is found, the program executes the statements following that **case**, and **all subsequent case** and **default** statements as well.
  - ✓ If no match is found with any of the **case** statements, only the statements following the **default** are executed.

switch statement

- Example 1

```c
#include <stdio.h>
int main( ){
    int i = 3 ;
    switch ( i ){
        case 1 :
            printf ( "I am in case 1 \n" ) ;
        case 2 :
            printf ( "I am in case 2 \n" ) ;
        case 3 :
            printf ( "I am in case 3 \n" ) ;
        default :
            printf ( "I am in default \n" ) ;
    }
    return 0;
}
```

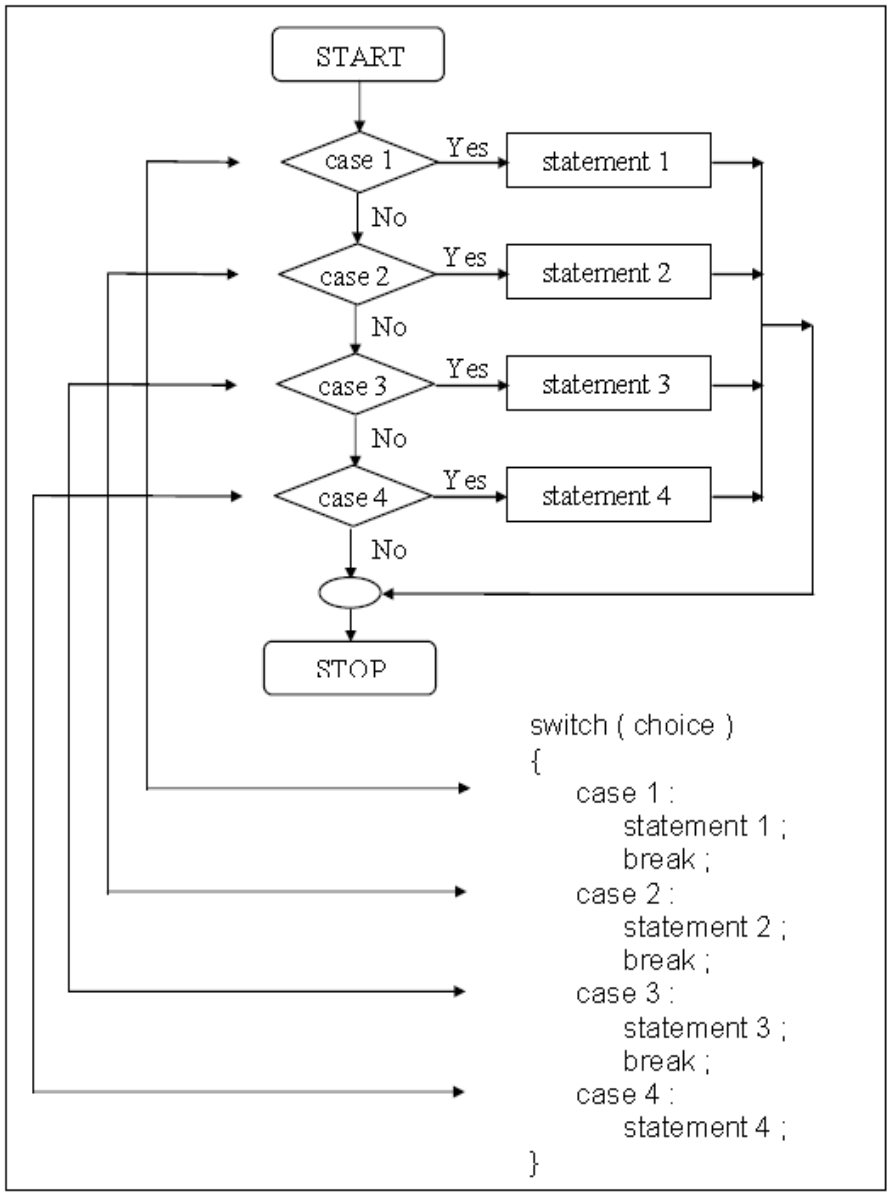
```
I am in case 3
I am in default
계속하려면 아무 키나 누르십시오 . . .
```

switch statement

- *break* statement

```c
#include <stdio.h>
main( ){
    int i = 3 ;
    switch ( i ){
        case 1 :
            printf ( "I am in case 1 \n" ) ;
            break;
        case 2 :
            printf ( "I am in case 2 \n" ) ;
            break;
        case 2+1 :
            printf ( "I am in case 3 \n" ) ;
            break;
        default :
            printf ( "I am in default \n" ) ;
    }
}
```

```
I am in case 3
계속하려면 아무 키나 누르십시오 . . .
```



switch statement

- Tip 1
  - ✓The earlier program that used **switch** may give you the wrong impression that you can use only cases arranged in ascending order, 1, 2, 3 and default.
  - ✓You can in fact put the cases in any order you please.

```c
#include <stdio.h>
int main( ){
    int i = 22 ;
    switch ( i )
    {
        case 121 :
            printf ( "I am in case 121 \n" ) ;
            break ;
        case 7 :
            printf ( "I am in case 7 \n" ) ;
            break ;
        case 22 :
            printf ( "I am in case 22 \n" ) ;
            break ;
        default :
            printf ( "I am in default \n" ) ;
    }
    return 0;
}
```

switch statement

- Tip 2
  - ✓ Allowed to use **char** values in **case** and **switch**

```
1  #include <stdio.h>
2  int main( ){
3      char c = 'x' ;
4      switch ( c ){
5          case 'v' :
6          printf ( "I am in case v \n" ) ;
7          break ;
8      case 'a' :
9          printf ( "I am in case a \n" ) ;
10         break ;
11     case 'x' :
12         printf ( "I am in case x \n" ) ;
13         break ;
14     default :
15         printf ( "I am in default \n" ) ;
16     }
17     return 0;
18 }
```

In fact here when we use ‘v’, ‘a’, ‘x’ they are actually **replaced by the ASCII values** (118, 97, 120) of these character constants.

Characters	ASCII Values
A – Z	65 – 90
a – z	97 – 122
0 – 9	48 – 57
special symbols	0 - 47, 58 - 64, 91 - 96, 123 - 127

```
I am in case x
계속하려면 아무 키나 누르십시오 . . .
```

switch statement

- Tip 3
  - ✓ At times we may want to execute a common set of statements for multiple **case**s

```c
#include <stdio.h>
int main( ){
    char c = 'x' ;
    switch ( c ){
        case 'v' :
        case 'V' :
            printf ( "I am in case v \n" ) ;
            break ;
        case 'a' :
        case 'A' :
            printf ( "I am in case a \n" ) ;
            break ;
        case 'x' :
        case 'X' :
            printf ( "I am in case x \n" ) ;
            break ;
        default :
            printf ( "I am in default \n" ) ;
    }
    return 0;
}
```

switch statement

- Tip 4
  - ✓Even if there are **multiple statements to be executed** in each **case** there is no need to enclose them within a pair of braces (unlike **if**, and **else**)

switch statement

- Tip 5
  - ✓Every statement in a **switch** must belong to some **case** or the other.
  - ✓If a statement doesn't belong to any **case** the **compiler won't report an error**. However, the statement would never get executed.

```
1  #include <stdio.h>
2  int main( ){
3      char c = 'x' ;
4      switch ( c ){
5          printf("test");
6          case 'v' :
7          case 'V' :
8              printf ( "I am in case v \n" ) ;
9              break ;
10         case 'x' :
11         case 'X' :
12             printf ( "I am in case x \n" ) ;
13             break ;
14         default :
15             printf ( "I am in default \n" ) ;
16     }
17     return 0;
18 }
```

```
I am in case x
계속하려면 아무 키나 누르십시오 . . .
```

switch statement

- Tip 6
  - If we have no **default** case, then the program simply falls through the entire **switch** and continues with the next instruction (if any,) that follows the closing brace of **switch**.
- Tip 7
  - The disadvantage of **switch** is that one cannot have a case in a **switch** which looks like: case i <= 20 :
  - All that we can have **after the case** is an **int** constant or a **char** constant or an expression that evaluates to one of these constants. Even a **float** is not allowed.
  - The advantage of **switch** over **if** is that it leads to a more structured program and the level of indentation is manageable, more so if there are multiple statements within each **case** of a **switch**.

switch statement

- Tip 8
  - ✓We can check the value of any expression in a **switch**. Thus the following **switch** statements are legal.
    - switch (i + j * k)
    - switch (23 + 45 % 4 * k)
    - switch (a < 4 && b > 7)
  - ✓Expressions can also be used in cases provided they are constant expressions. Thus **case 3 + 7** is correct, however, **case a + b** is incorrect.

switch statement

```c
#include <stdio.h>
int main( ){
    int i = 3 ;
    int k = 1;
    switch ( i ){
        case 1 :
            printf ( "I am in case 1 \n" ) ;
            break;
        case 2 :
            printf ( "I am in case 2 \n" ) ;
            break;
        case 2+k :
            printf ( "I am in case 3 \n" ) ;
            break;
        default :
            printf ( "I am in default \n" ) ;
    }
    return 0;
}
```

switch statement

- Tip 9
  - ✓The **break** statement when used in a **switch** takes the control outside the **switch**.
- Tip 10
  - ✓In principle, a **switch** may occur within another, but in practice it is rarely done. Such statements would be called nested **switch** statements.

switch statement

- There are some things that you simply cannot do with a **switch**. These are:
  - ✓ A float expression cannot be tested using a **switch**
  - ✓ Cases can never have variable expressions (for example it is wrong to say **case a +3 :**)
  - ✓ Multiple cases cannot use same expressions. Thus the following **switch** is illegal:

```c
#include <stdio.h>
int main( ){
    int i = 22 ;
    switch ( i )
    {
        case 121 :
            printf ( "I am in case 121 \n" ) ;
            break ;
        case 22 :
            printf ( "I am in case 7 \n" ) ;
            break ;
        case 22 :
            printf ( "I am in case 22 \n" ) ;
            break ;
        default :
            printf ( "I am in default \n" ) ;
    }
    return 0;
}
```

Summary

- When we need to choose one among number of alternatives, a ***switch*** statement is used.
- The ***switch*** keyword is followed by an integer or an expression that evaluates to an integer.
- The ***case*** keyword is followed by an integer or a character constant.
- The control falls through all the cases unless the ***break*** statement is given.

실습

- Switch 문을 활용하여 입력된 숫자가 짝수인지 홀수인지 판단하는 프로그램을 작성하시오

Exercise

- What would be the output of the following programs:

(a)
```
main()
{
    char suite = 3;
    switch (suite)
    {
        case 1 :
            printf ("\nDiamond");
        case 2 :
            printf ("\nSpade");
        default :
            printf ("\nHeart");
    }
    printf ("\nI thought one wears a suite");
}
```

(b)
```
main()
{
    int c = 3;

    switch (c)
    {
        case 'v' :
            printf ("I am in case v \n");
            break;
        case 3 :
            printf ("I am in case 3 \n");
            break;
        case 12 :
            printf ("I am in case 12 \n");
            break;
        default :
            printf ("I am in default \n");
    }

}
```

Exercise

- What would be the output of the following programs:

(c)
```
main()
{
    int k, j = 2 ;
    switch ( k = j + 1 )
    {
        case 0 :
            printf ( "\nTailor") ;
        case 1 :
            printf ( "\nTutor") ;
        case 2 :
            printf ( "\nTramp") ;
        default :
            printf ( "\nPure Simple Egghead!" ) ;
    }
}
```

(d)
```
main()
{
    int i = 0 ;
    switch ( i )
    {
        case 0 :
            printf ( "\nCustomers are dicey" ) ;
        case 1 :
            printf ( "\nMarkets are pricey" ) ;
        case 2 :
            printf ( "\nInvestors are moody" ) ;
        case 3 :
            printf ( "\nAt least employees are good" ) ;
    }
}
```

Exercise

- What would be the output of the following programs:

(e)
```
main()
{
    int k;
    float j = 2.0;

     switch ( k = j + 1 )
     {
          case 3 :
               printf ( "\nTrapped" ) ;
               break ;
          default :
               printf ( "\nCaught!" ) ;
     }
}
```

(f)
```
main()
{
    int ch = 'a' + 'b';
    switch ( ch )
    {
         case 'a' :
         case 'b' :
              printf ( "\nYou entered b" ) ;
         case 'A' :
              printf ( "\na as in ashar" ) ;
         case 'b' + 'a' :
              printf ( "\nYou entered a and b" ) ;
    }
}
```

Exercise

- What would be the output of the following programs:

```
(g)   main( )
      {
          int  i = 1 ;
          switch ( i - 2 )
          {
               case -1 :
                    printf ( "\nFeeding fish" ) ;
               case 0 :
                    printf ( "\nWeeding grass" ) ;
               case 1 :
                    printf ( "\nmending roof" ) ;
               default :
                    printf ( "\nJust to survive" ) ;

            }
       }
```

HW #4

1. Print out the errors, if any, in the following programs:

(a)
```
main( )
{
    int suite = 1 ;
    switch ( suite ) ;
    {
        case 0 ;
            printf ( "\nClub" ) ;
        case 1 ;
            printf ( "\nDiamond" ) ;
    }
}
```

(b)
```
main( )
{
    int temp ;
    scanf ( "%d", &temp ) ;
    switch ( temp )
    {
        case ( temp <= 20 ) :
            printf ( "\nOoooooohhhh! Damn cool!" ) ;
        case ( temp > 20 && temp <= 30 ) :
            printf ( "\nRain rain here again!" ) ;
        case ( temp > 30 && temp <= 40 ) :
            printf ( "\nWish I am on Everest" ) ;
        default :
            printf ( "\nGood old nagpur weather" ) ;
    }
}
```

HW #4

1. Print out the errors, if any, in the following programs:

(c)
```
main()
{
    float a = 3.5 ;
    switch ( a )

    {
        case 0.5 :
            printf ( "\nThe art of C" ) ;
            break ;
        case 1.5 :
            printf ( "\nThe spirit of C" ) ;
            break ;
        case 2.5 :
            printf ( "\nSee through C" ) ;
            break ;
        case 3.5 :
            printf ( "\nSimply c" ) ;
    }
}
```

(d)
```
main()
{
    int a = 3, b = 4, c ;
    c = b – a ;
    switch ( c )
    {
        case 1 || 2 :
            printf ( "God give me an opportunity to change things" ) ;
            break ;

        case a || b :
            printf ( "God give me an opportunity to run my show" ) ;
            break ;
    }
}
```

HW #4

2. Write a program that converts uppercase characters to lowercase and passed non-uppercase characters unchanged.
   - ✓ char data type
   - ✓ ASCII table