CS1100 - Lecture 13

Instructor : Jasine Babu

Teaching Assistants : Nikhila K N, Veena Prabhakaran

Switch statement

Switch is a control statement, the format for writing a switch statement in a generic way is as follows.

```
switch(expression)
{
   case constant_expression 1: Block 1
   case constant_expression 2: Block 2
   .
   .
   .
   case constant_expression k: Block k
   default: Block k+1
}
Block k+2
```

In the above format, `switch` is a keyword just like `if, while`. Then we write in brackets an expression, which has an integer value. In a switch statement, the expression should be always an integer-valued or something that is of the category of an integer. There are some different types of integers, such as *long integers*, *short integers*, a *character* is also are considered as integers. Any of them can come in the value of the expression. But a *float* or *string* can not come as a value of the expression. We can not switch based on a *string* or *float* value.

In the switch statement `case` is a keyword, each `case` has a constant expression. When the program control reaches the switch statement, first the expression in the switch is evaluated. The expression has an integer value, this integer value is compared with the values of each case. When the expression value is matched with the value of any expression in the case, the every Block following that case will executes. If the expression value does not match any of the expression in the case. Then the Block in the `default` will be executed.

Switch statement with break

Use `break` in the switch statement is the most useful way of using the switch. The format for writing a switch statement with the break is shown in below.

```
switch(expression)
{
   case constant_expression 1: Block 1
                               break;
   case constant_expression 2: Block 2
                               break;
   .
   .
   .
   case constant_expression k: Block k
                               break;
   default: Block k+1
            break;
}
Block k+2
```

In this form, the switch expression is evaluated and the value is compared with the values of each case. Suppose the expression value is matched with the value of the constant_expression 2 in the case. Then the Block 2 is executed. After executing the Block 2, it has a `break` statement. This means, the Blocks after the Block 2 will be skipped and program control comes out from the switch.

Suppose we have a many *else if* as follows.

```
if(expression==constant_expression 1)
{
    Block 1
}
else if(expression==constant_expression 2)
{
    Block 2
}
else if(expression==constant_expression 3)
{
    Block 3
}
.
.
.
else if(expression==constant_expression k)
{
    Block k
}
else
{
    Block k+1
}
Block k+2
```

The expression in the each `if` has an integer value. Use switch with break is the better way to do this.

The following program illustrates the use of switch statement. This is the program for finding grade based on marks. Suppose we are following absolute grading. If the mark is above 90, it will say grade `S`. Marks above from 80-89, it will say grade `A`. If the mark is below 40, the grade is `F`.

```
#include<stdio.h>
void main()
{
   int marks;
   char grade;

   printf("\nEnter the marks out of 100\n");
   scanf("%d",&marks);

   switch(marks/10)
   {
       case 0:
       case 1:
       case 2:
       case 3: grade='F';
               break;
       case 4: grade='E';
               break;
       case 5: grade='D';
               break;
       case 6: grade='C';
               break;
       case 7: grade='B';
               break;
       case 8: grade='A';
               break;
       case 9:
       case 10: grade='S';
                break;
   }
   printf("Grade is : %c\n", grade);
}
```

This program takes the input from the user. Suppose we give the marks=90 as input. After executing the program, the output `Grade is :  S` will be displayed.