Lecture 3
Programming Essentials in C

Control FLOW Statements

- C language has decision making capabilities and supports controlling of statements.
- C supports following decision making statements:
  1. IF
  2.IF-ELSE and its various form
  3. Switch
  4. Conditional Operator

If statement

- The if statement is a powerful decision making statement.
- The if statement is used to control the flow of execution of statements.
- It is a two way decision statement which is used in conjunction with an expression.

Different forms of If statement

- Simple If statement.
- If…..else statement.
- Nested if…..else statement.
- Else if ladder.

Simple if statement

- If the if expression evaluates to true, the block following the if statement or statements are executed.

- The general form of a simple if statement is :

```
if (test-expression)
   {
      statement-block;
   }
  statement-x;
```

If….Else statement

- The general form of if……else statements is :

```
if (expression)
        statement;
else
        statement;
```

- IF the if expression evaluates to true, the block following the if statement or statements are executed.

- The else statement is optional. It is used only if a statement or a sequence of statements are to be executed in case the if expression evaluates to false.

Else if ladder

- The general form of else if ladder is:

```
if (expression)
        statement;
else if (expression)
        statement;
else if (expression)
        statement;
          .
          .
          .
else
        statement;
```

- The if – else – if statement is also known as the if-else-if ladder or the if-else-if staircase.
- The conditions are evaluated from the top downwards.

Nested if…else statement

- The general form of nested if…else statement is:

```
if (test-expression 1)
 {
      if (test-expression 2)
      {
                statement 1;
       }
     else
       {
              statement 2;
        }
  }
else
 {
       statement 3;
 }
Statement x;
```

Switch – case statement

- The switch-case control statement is a multi-way decision maker that tests the value of an expression against a list of integers or character constants.
- When a match is found, the statements associated with that constant are executed.
- The syntax of switch-case is:-

```
switch (expression)
{
        case constant1:
                statement sequence
                break;
        case constant2:
                statement sequence
                break;
        case constant3:
                statement sequence
                break;
        .
        .
        .
        default:
                statement sequence
}
```

Jumping Statement

- Break
- Continue
- Goto

Break statement

- When the keyword break is encountered inside any c loop, control automatically passes to the first statement after the loop.
- A break is usually associated with an if.
- The general syntax of break statement is:

```
For(; ;)
{
      --------
      --------
     if (error)
      break;
      -------
 }
    --------
```

Continue statement

- In some programming situations we want to take the control to the beginning of the loop, by passing the statements inside the loop which have not yet been executed. The Keyword continue allows us to do this.
- When the keyword continue is encountered inside any C loop, control automatically passes to the beginning of the loop.
- A continue is usually associated with an **if.**

```
#include<stdio.h>
void main()
{
int i,j;
for(i=1;i<=2;i++)

{
for(j=1;j<=2;j++)
{
if(i==j)
continue;
printf(“%d%d”,i,j);
}
}    }
```

Output 1 2

2 1

Go to statement

- C supports the go to statement to branch unconditionally from one point to another in the program.
- It requires a label in order to identify the place where branch is to be made.
- The general syntax of go to statement is:

```
go to label;
        --------
        --------
      label:
        statement x ;
```