SLIIT

Discover Your Future

[redacted]

IT1010 – Introduction to Programming

Lecture 4 – Selection Statements in C / Character Handling

SLIIT
FACULTY OF COMPUTING

Objectives

- At the end of the Lecture students should be able to

  - use the different types of selection statements to select actions (if, if …else, nested if, conditional operator, switch).

  - use getchar( ) function to read characters from the key board

Decision Making using If statement

if statement performs an action if a **condition** is **true**. Conditions in if statements are formed by using the equality operators and relational operators

```
// using if statement
#include <stdio.h>
int main(void)
{

        int no1, no2;
        printf( “%s”, “Enter two integers : ” );
        scanf( “%d%d”, &no1, &no2);        // read two integers
        if ( no1 == no2 )    // checking equal
                printf( “%d is equal to %d\n” , no1, no2 );

        if ( no1 != no2 ) // checking not equal
                printf( “%d is not equal to %d\n” , no1, no2 );


        return 0;
 } // end of main function
```

output

```
Enter two integers :3  3
3 is equal to 3
```

```
Enter two integers :5  3
5 is not equal to 3
```

if statement cont…

```
if ( no1 == no2 )
{
     printf( "%d is equal to %d\n" , no1, no2 );
     printf( "%s", "Numbers are same " );
}
```

- To include several statements in the body of an if, enclose the set of statements in braces ({ and })
- A left brace ({) begins the body of each if statement
- A corresponding right brace (}) ends each if statement body
- Any number of statements can be placed in the body of an if statement
- A set of statements contained within a pair of braces is called a **compound statement** or a **block**

Exercise 01

Write a program in C to read two integer numbers from the keyboard and display the largest number.

if …. else Statement

- if ….. else statement performs an action if a condition is true and performs a different action if the condition is false

```
/* printing pass or fail using if .. else  statement */
#include <stdio.h>
int main(void)
{
       int mark;

       printf( "Enter marks : " );
       scanf( "%d", &mark );       // read marks

       if ( mark >= 60 )    // check whether mark greater than or equal to  60
       {
               printf( "%s", "Passed" );
       }
       else
       {
               printf( "%s", "Failed " );
               printf( "You must take this again \n" );
       }

        return 0;
 }
```

Conditional Operator

- Conditional operator(?:) is related to the if ….else statement.
- It takes three operands. First operand is a condition. Second is the value if the condition is true. Third is the value if the condition is false.

Example

```
mark >= 60 ? printf( “Passed\n” ) : printf( “Failed\n” );
```

Above statement is same as,

```
if ( mark >= 60 )
    printf( “Passed” );
else
    printf( “Failed” );
```

Nested if…. else statements

- Nested if … else statements handle multiple cases by placing if …else statements inside if …else statements.

```
/* printing grade using nested if .. else  statement */
#include <stdio.h>
int main(void)
{
        int mark;

        printf(“%s”, “Enter marks : ”);
        scanf(“%d”, &mark);       // read marks

        if ( mark >= 80 )
             printf( “%s”, “Grade A” );
        else if ( mark >= 50 )
             printf( “%s”, “Grade B ” );
        else if ( mark >= 40 )
             printf( “%s”, “Grade C ” );
         else
              printf( “%s”, “Grade F ” );

        return 0;
}
```

Switch Statement

- The switch statement is an alternative to the nested if-else statement provided the expressions can be written as:

  (variable == value)

- The switch statement consists of a series of case labels
- Multiple statements can be executed for a given condition and break statement terminates the execution of the condition

Switch Statement - Example

Syntax

```
switch (variable)
{
   case c1:  any_number_of_statements;
             break;

   case c2: any_number_of_statements;
             break;
              …

   default: any_number_of_statements;
}
```

Example

```
#include <stdio.h>
int main(void)
{
        int score;

        printf( “%s”, “Enter score  : ” );
        scanf( “%d”, &score );      // read score

        switch ( score )
        {
             case 3 : printf( “ Congratulations\n” );
                      printf( “ Gold Winner\n” );
                       break;
             case 2 : printf( “ Silver Winner\n” );
                       break;
             case 1 : printf( “ Bronze Winner\n” );
                      break;
             default : printf( “ Invalid Score\n” );

        }
        return 0;
}  // end of main function
```

char data type

- Characters are normally stored in variable type char
- Characters can be stored in any integer type variable too
- Characters can be treated as either an *integer* or a *character*
- **getchar** function reads one character from the keyboard
- Characters can be read with **scanf**

by using the conversion specifier %c

```
// reading a character and print messages appropriately
#include <stdio.h>
int main(void)
{
          int grade;

          printf( “%s”, “Enter grade  : ” );
          grade = getchar();     // read a character

          switch( grade )
          {
                case ‘A’ :  printf( “%s”, “Excellent” );
                            break;
                case ‘B’ :  printf( “%s”, “Good” );
                            break;
                 …………..
                 …………..


          return 0;
}       // end of main function
```

char data type cont…

- Many computers today use ASCII(American Standard Code for Information Interchange) character set

Example:

```
printf( “The character (%c) has the value %d.\n”, ‘a’, ‘a’ );
```

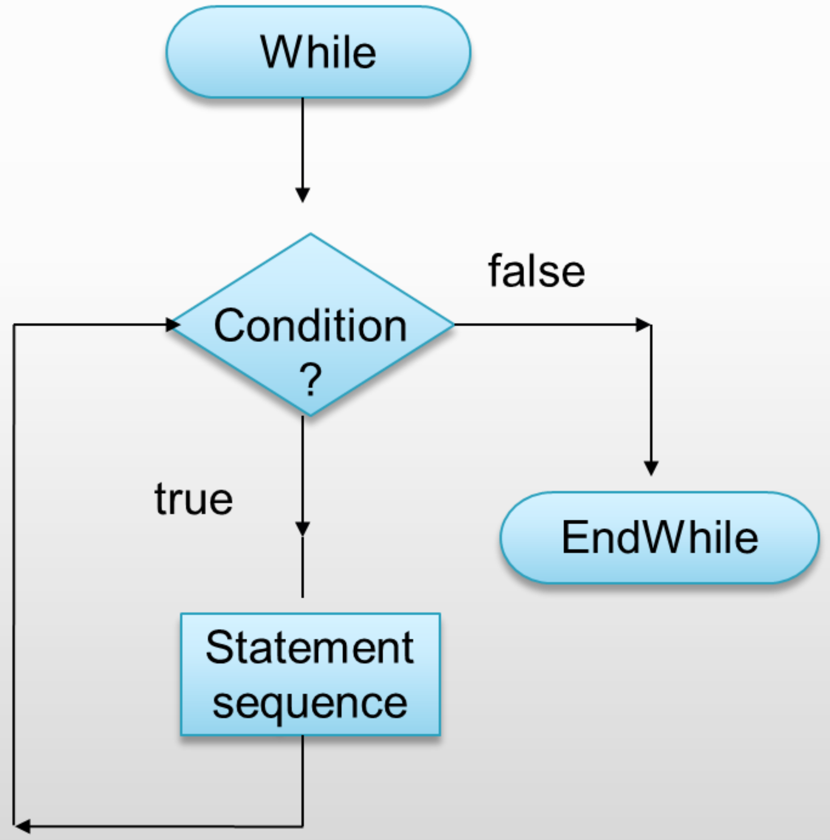
Output :

```
 The character (a) has the value 97.
```

- Conversion specifier %c and %d can be used to print character ‘a’ and its integer value
- 97 is the numerical representation of character ‘a’ in the computer.

Iteration

- Certain steps may need to be repeated while, or until, a certain condition is true.
- We call it as a Loop

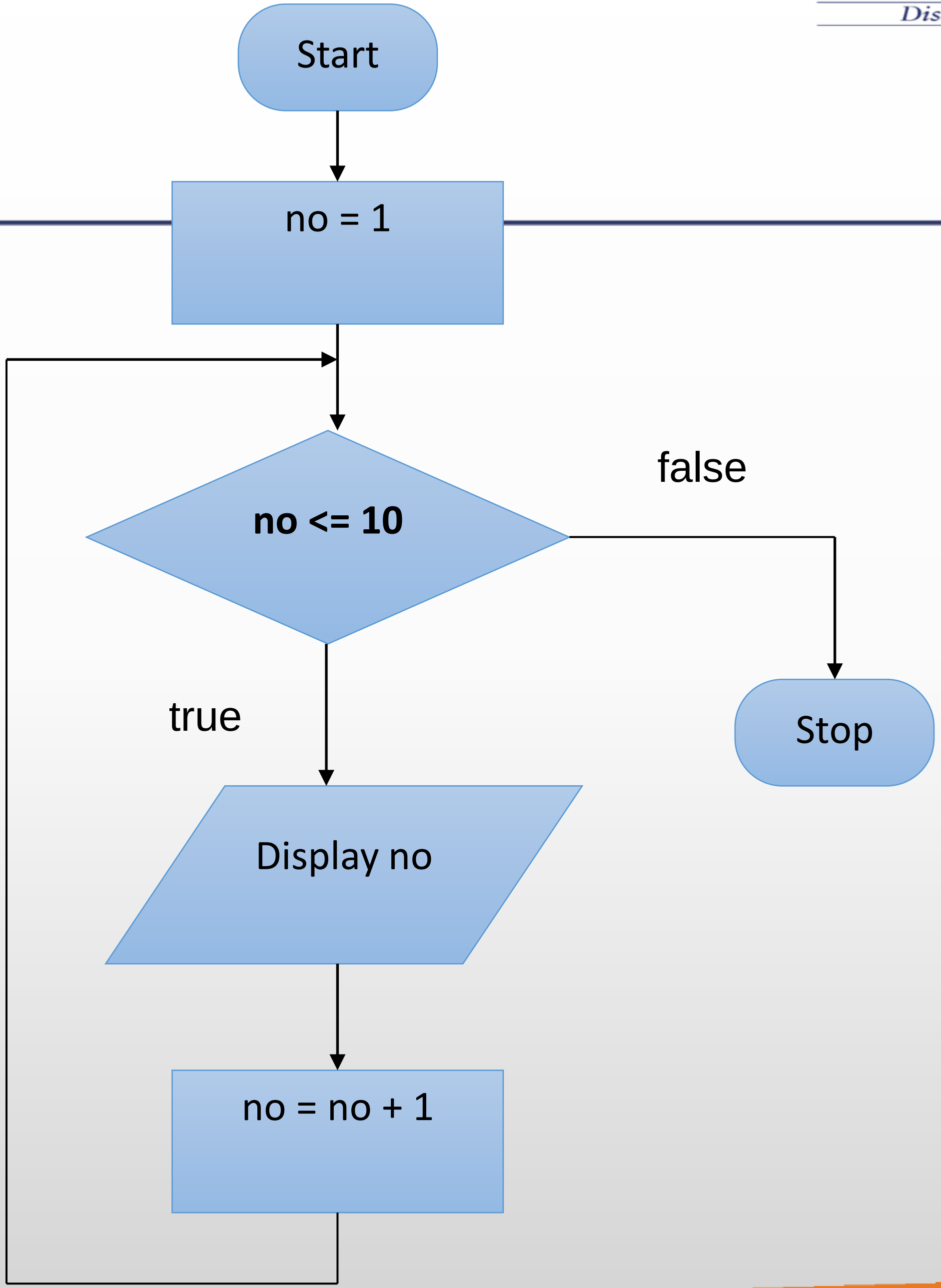


Iteration

- Section(s) of an algorithm are repeated over and over (obviously these loops must eventually terminate)
- This is achieved by a test of whether a condition is true or false
- In a while loop we continue to repeat something **while a condition is true – we terminate the loop when it is false**

Example

- Draw a flowchart to represent an algorithm to display the numbers 1, 2, 3, 4, 5, ….., 10



Exercise 02

- Draw a flowchart to find the sum of 10 numbers entered through the keyboard.

Exercise 03

- Draw a flowchart to find the average of 10 numbers entered through the keyboard.

Summary

- If statement
- If .. Else statement
- Conditional operator
- Nested selection
- Switch statement
- getchar ( )
- Iteration