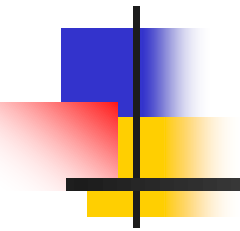

Condition

Session 5

Objectives

- Explain the Selection Construct
  - If Statement
  - If – else statement
  - Multi if statement
  - Nested if statement
- Switch statement

Conditional Statement

- Conditional statements enable us to change the flow of the program

- A conditional statement evaluates to either a true or a false value

Example :

To find whether a number is even or odd we proceed as follows :

1. **Accept a number**
2. **Find the remainder by dividing the number by 2**
3. **If the remainder is zero, the number is "EVEN"**
4. **Or if the remainder is not zero the number is "ODD"**

Selection Constructs

C supports two types of selection statements

The if statement

The switch statement

The if statement-1

Syntax:

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

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

The if statement-2

Program to display the values based on a condition

```
#include <stdio.h>
void main()
{
   int x, y;
   char a = 'y';
   x = y = 0;
   if (a == 'y')
   {
      x += 5;
      printf("The numbers are %d and \t%d", x, y);
   }
}
```

Example

The if – else statement-1

Syntax:

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

The if – else statement-2

- If the if expression evaluates to true, the block following the if statement or statements are executed
- If the if expression does not evaluate to true then the statements following the else expression take over control
- 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

The if – else statement -3

Program to display whether a number is Even or Odd

```
#include <stdio.h>
void main()
{
   int num , res ;

   printf("Enter a number :");
   scanf("%d",&num);
   res = num % 2;
   if (res == 0)
       printf("Then number is Even");
   else
       printf("The number is Odd");
}
```

Example

The if–else–if statement-1

Syntax:

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

The if–else–if statement-2

- 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

The if–else–if statement-3

Program to display a message based on a value

```
#include <stdio.h>
   main()
   {
      int x;
      x = 0;
      clrscr ();
      printf("Enter Choice (1 - 3) : ");
      scanf("%d", &x);
      if (x == 1)
         printf ("\nChoice is 1");
      else if ( x == 2)
         printf ("\nChoice is 2");
      else if ( x == 3)
         printf ("\nChoice is 3");
      else
         printf ("\nInvalid Choice ");
   }
```

Example

Nested if-1

- The nested **if** is an **if** statement, which is placed within another **if** or **else**

- In C, an **else** statement always refers to the nearest **if** statement that is within the same block as the **else** statement and is not already associated with an **if**

Nested if-2

Syntax:

```
if (exp1)
{
     if (exp2) statement1;
     if (exp3) statement2;
     else statement3;          /*with if (exp3) */
}
else statement4;               /* with if (exp1) */
```

- Note that the inner else is associated with **if(exp3)**
- According to ANSI standards, a compiler should support at least 15 levels of nesting

Nested if-3

```
#include <stdio.h>
void main ()
 {
    int x, y;
    x = y = 0;
    clrscr ();
    printf ("Enter Choice (1 - 3) : ");
    scanf ("%d", &x);
    if (x == 1)
    {
       printf("\nEnter value for y (1 - 5) : ");
       scanf ("%d", &y);
       if (y <= 5)
              printf("\nThe value for y is : %d", y);
       else
              printf("\nThe value of y exceeds 5 ");
    }
    else
       printf ("\nChoice entered was not 1");
}
```

Example

The switch statement-1

- The **switch** 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 switch statement-2

Syntax:

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

The switch statement-3

Program to check whether the entered lowercase character is vowel or 'z' or a consonant

```
#include <stdio.h>
   main ()
   {
      char ch;
      clrscr ();

      printf ("\nEnter a lower cased alphabet (a - z) : ");
      scanf("%c", &ch);
```

Example

contd.....

The switch statement-4

```
if (ch < 'a' || ch > 'z')
        printf("\nCharacter not a lower cased alphabet");
else
        switch (ch)
        {
                case 'a' :
                case 'e' :
                case 'i' :
                case 'o' :
                case 'u' :
                        printf("\nCharacter is a vowel");
                        break;
                case 'z' :
                        printf ("\nLast Alphabet (z) was entered");
                        break;
                default :
                        printf("\nCharacter is a consonant");
                        break;
                        }
        }
```

Example