***Topic 3:- Decision Control Statements*.**

Decision control Statement:- *The decision control statements are the decision making statements that decides the order of execution of statements based on the conditions. In the decision making statements the programmer specify which conditions are to be executed or tested with the statements to be executed if the condition is true or false.*

I. *If statement*
II. *else if statement*
III. *nested if else statement*
IV. *switch statement*

I. *if statement :-* *If statement is used to control the program , if the condition is true then the statements will be executed or else not. This statement is the simple and easy decision control statement.*

Syntax:-

```
if(condition)
{
Statement 1;
Statement 2;
……..
Statement n;
}
```

Example:-

Question 1:-Write a program to age, if your age 50 above then *you are old* using if condition.

Solution:-

```
#include<iostream>
using namespace std;
```

```
int main()
{
int m;
cout<<"Enter any one age";
cin>>m;
if(m>=50&&m<=100)
{
cout<<"you are old.";
}
return 0;
}
```

Output:-

Enter any one age 85 ***// first time run you enter 85***

you are old.

Enter any one age 101 ***// second time run then you enter 101***

Nothing

Enter any one age 12 ***// third time run then you enter 12***

Nothing

II. ***if else Statement:-*** If you have a condition and you want to execute a block of code if condition is true and execute another piece of code if the same condition is false that time we use if else statement.

***Syntax*:-**

```
if(condition)
{
Statement 1;
Statement 2;
……..
```

```
Statement n;
}
else
{
Statement 1;
Statement 2;
……..
Statement n;
}
```

Example:-

Question 1:-Write a program to age, if your age 50 above then *you are old* and your age 50 less than you are young using if else condition.

Solution:-

```
#include<iostream>
using namespace std;
int main()
{
int m;
cout<<"Enter any one age";
cin>>m;
if(m>=50&&m<=100)
{
cout<<"you are old.";
}
else
```

```
{
cout<<"you are young.";
}
return 0;
}
```

***Output*:-**

```
Enter any one age 85      // first time run you enter 85
you are old.
Enter any one age 40      // second time run then you enter 40
you are young
Enter any one age 12      // third time run then you enter 12
you are young
```

Note:-If you are enter (0 and – value and above 100 value) than you will run than you will find <u>you are young</u> that is wrong .So we will discuss a new topic. New topic is nested if else.

III. ***<u>Nested if else Statement:-</u>*** Nested if else(if-else-if) statement is used when we need to check multiple conditions.

***Syntax*:-**

```
if(condition)
{
Statement 1;
Statement 2;
……..
Statement n;
}
else if(condition)
```

```
{
Statement 1;
Statement 2;
……..
Statement n;
}
else if(condition)
{
Statement 1;
Statement 2;
……..
Statement n;
}
……
else
{
Statement 1;
Statement 2;
……..
Statement n;
}
```

Example:-

Question 1:-Write a program to age, if your age 50 above then *you are old* and your age 50 less than you are young

Otherwise wrong value using nested if else condition.

Solution:-

```
#include<iostream>
using namespace std;
int main()
{
int m;
cout<<"Enter any one age";
cin>>m;
if(m>=50&&m<=100)
{
cout<<"you are old.";
}
else if(m>0&&m<50)
{
cout<<”you are young.”;
}

else
{
cout<<”wrong value.”;
}
return 0;
}
```

Output:-

Enter any one age 85 *// first time run you enter 85*

you are old.

Enter any one age 40 *// second time run then you enter 40*

you are young

Enter any one age 12 *// third time run then you enter 12*

you are young

Enter any one age 0 or - value or 100 above *// third time run then you enter 0 or - value or 100 above*

Wrong value

(IV). Switch statement:-This statement use to in c++ two types.

1. *Switch case.*
2. *Switch case break.*

1. Switch case:- It is used when we have multiple conditions and we need to perform different action based on the condition. When we have multiple conditions and we need to execute a block of statements when a particular condition is satisfied. In such case either we can use lengthy if-else-if statement or switch case. The problem with lengthy if-else-if is that it becomes complex when we have several conditions. The switch case is a clean and efficient method of handling such scenarios.

Syntax:-

```
switch (variable or an integer expression)
    {
      case constant:
          Statement 1;
          Statement 2;
          ....
          Statement n;
      case constant:
          Statement 1;
```

```
            Statement 2;
            ….
            Statement n;
        default:
            Statement 1;
            Statement 2;
            ….
            Statement n;
    }
```

Example:-

```
#include<iostream>
using namespace std;
int main()
{
int a;
cout<<"enter one number";
cin>>a;
switch(a)
{
    case 1:
        cout<<"Case first"<<a<<”\n”;
    case 2:
        cout<<"Case second"<<a<<”\n”;
    default:
        cout<<"wrong number";
}
return 0;
```

```
}
```

Output:-

Enter one number 1 *// first time run you enter 1*

Case first

wrong number

Enter one number 2 *// second time run you enter 2*

Case second

wrong number

Enter one number 3 or above 3 *// third time run you enter 3 or above 3*

wrong number

Note:- ***Above program we can see that default value gave you all time, this is wrong. So we will discuss switch case break statement.***

2. ***switch case break*:-** it is used when you want your program-flow to come out of the switch body. Whenever a break statement is encountered in the switch body, the execution flow would directly come out of the switch, ignoring rest of the cases. This is why you must end each case block with the break statement .

***Syntax*:-**

```
switch (variable or an integer expression)
    {
       case constant:
    Statement 1;
          Statement 2;
          ....
          Statement n;
    break;
```

```
    case constant:
Statement 1;
        Statement 2;
        ….
        Statement n;
break;
default:
 Statement 1;
        Statement 2;
        ….
        Statement n;
 break;
   }
```

Example:- ***Write a program to put any alphabet and find vowels and consonant***.

Solution:-

```
#include<iostream>
using namespace std;
int main()
{
char ab;
cout<<"enter any one alphabet";
cin>>ab;
```

```
switch(ab)
{
	case 'a'|'A':
		cout<<"vowels";
	break;
	case 'e' | 'E':
		cout<<"vowels";
	break;

	case 'I' | 'i':
		cout<<"vowels";
	break;
	case 'O' | 'o':
		cout<<"vowels";
	break;
	case 'u' | 'U':
		cout<<"vowels";
	break;
	default:
		cout<<"consonant";
	break;
}
return 0;
}
```

Output:-

enter any one alphabet a or A ***// first time run you enter a or A***

vowels

enter any one alphabet e or E ***// second time run you enter e or E***

vowels

enter any one alphabet i or I ***// third time run you enter i or I***

vowels

enter any one alphabet o or O ***// four time run you enter o or O***

vowels

enter any one alphabet u or U ***// five time run you enter u or U***

vowels

enter any one alphabet apart from vowels ***// all time run you enter apart from vowels***

consonant

Created by:-Ajay kumar verma