Jump Statement

This statement is used to transfer the control in the program. Mainly we use this statements in the loop to

- Tterminate the execution of loop
- Exiting a loop
- In between we can skip from the loop

We can achieve this by using break, exit, goto and continue statement

Break statement: We can use this statement to terminate the flow of execution we can use this

- To terminate a case in the switch statement
- To transfer the control to out side the loop

When the break statement is encountered inside the loop, the control is transferred outside the loop

Syntax: **break;**

```
Example:  for (i=0; i<20; i++)
        {
          cout<<i;
          if(i= =5)
          break;
}
cout<<"loop is terminated ";
```

The above program segment prints the number from 0 to 5 on the screen. Then terminated from the loop and we get the output on screen as loop is terminated.

The continue statement: This statement used to transfer the control to the beginning of the loop when specified condition is occurred.

The general syntax: **continue;**

```
Example:  for (i=0; i<6; i++)
        {
          cout<<i;
          if(i= =3)
          continue;
}
cout<<"loop is terminated ";
```

The above program segment prints the value 0 1 2 4 5 on the screen by skipping 3. When the value of i=6 we get the output as loop is terminated.

The goto statement: The goto statement is a simple statement used to transfer control from one point of a program to another point of program without any condition. Due to that goto is called as undisciplined statement.

syntax:

```
goto label:statement1;
               statement2;
label: statement3;
```

where label is user defined identifier , it may appear either before or after goto statement. There is no need of declaration of label. The syntax of label requires **colon(:)**

Example:

```
i=0;
san: cout<<i;
i++;
if(i<5)
goto san;
```

The above programming segment will print the value 0 1 2 3 4

The exit() function : The exit() is a standard library function readily available with c++ compiler . Its purpose is to terminate the execution of the program itself. Its definition is present under header file **stdlib.h**

Syntax: **exit ();**

Example:

```
for(i=2;i<=n/2;i++)
{
if(n%i= = 0)
{
cout<<n<<”is not a prime number”;
exit(0);
}
cout<<n<<”is a prime”;
}
```

In the above programming segment if the number is divisible by any number from 2 to half of the number. The program displays a message number is not prime and exit the program.