Dr.SNS RAJALAKSHMI COLLEGE OF ARTS AND SCIENCE, (AUTONOMOUS)

COIMBATORE-641049

Accredited by NAAC (Cycle III) with “A+” Grade Recognized by UGC, Approved by AICTE, New Delhi andAffiliated to Bharathiar University, Coimbatore.

DEPARTMENT OF COMPUTER APPLICATIONS

Course Code / Course Name: **23UCU401 /Programming in C**

YEAR:	**2023-2024**
CLASS:	**I BCA “A”**
STAFF NAME:	**Dr.A.DEVI**

Control Statement

Generally C program statement is executed in a order in which they appearin the program. But sometimes we use decision making condition for execution only a part of program, that is called control statement. Control statement defined how the control is transferred from one part to the other part of the program. Thereare several control statement like if...else, switch, while, do. while, for loop,
break, continue, goto etc.

Loops in C

Loop:-it is a block of statement that performs set of instructions. In loops
Repeating particular portion of the program either a specified number of time oruntil a particular no of condition is being satisfied.
There are three types of loops in c
1.While loop 2.do while loop 3.for loop While loop
Syntax:-

```
while(conditio
n)

{

Statement 1;

Statement 2;


O      while(test
       condition)
```

The test condition may be any expression .when we want to do something a fixedno of times but not known about the number of iteration, in a program then whileloop is used.
Here first condition is checked if, it is true body of the loop is executed else, If condition is false control will be come out of loop.
Example:-

```
/* wap to print 5 times welcome to C” */

#include<stdio.h>

void main()

{

int p=1;

While(p<=5)

{

printf(“Welcome to C\n”);

P=p+1;

}

}
```

Output: Welcome to C
Welcome to C
Welcome to C Welcome
to C Welcome to C

So as long as condition remains true statements within the body of while loop willget executed repeatedly.

do while loop

This (do while loop) statement is also used for looping. The body of this loop maycontain single statement or block of statement. The syntax for writing this statement is:

Syntax:-

```
Do

{

Statement;

}

while(condition);
```

Example:-

```
#include<stdio.h>
void main()
{
int   X=4;
 do
{
 Printf("%d",X);
X=X+1;
  }whie(X<=10);
    Printf(" ");
}
```

Output: 4 5 6 7 8 9 10

Here firstly statement inside body is executed then condition is checked. If the condition is true again body of loop is executed and this process continue until thecondition becomes false. Unlike while loop semicolon is placed at the end of while.

There is minor difference between while and do while loop, while loop test thecondition before executing any of the statement of loop. Whereas do while looptest condition after having executed the statement at least one within the loop.

If initial condition is false while loop would not executed it's statement on other hand do while

loop executed it’s statement at least once even If condition fails forfirst time. It means do while loop always executes at least once. **Notes:**
Do while loop used rarely when we want to execute a loop at least once.

for loop

In a program, for loop is generally used when number of iteration are known in advance. The body of the loop can be single statement or multiple statements. Itssyntax for writing is:

Syntax:-

```
for(exp1;exp2;exp3)

{

Statement;

}
```

Or

```
for(initialized counter; test counter; update counter)

{

Statement;

}
```

Here exp1 is an initialization expression, exp2 is test expression or condition andexp3 is an update expression. Expression 1 is executed only once when loop started and used to initialize the loop variables. Condition expression generally uses relational and logical operators. And updation part executed only when afterbody of the loop is executed.

Example:-

```
void main()
{
int                    i;
for(i=1;i<10;i++)
{
 Printf(“ %d ”, i);
```

```
  }

}
```

Output:-1 2 3   4 5 6 7 8 9

Nesting of loop

When a loop written inside the body of another loop then, it is known as nesting ofloop. Any type of loop can be nested in any type such as while, do while, for. For example nesting of for loop can be represented as :

```
void main()
{
int   i,j;   for(i=0;i<2;i++)
for(j=0;j<5;             j++)
printf("%d %d", i, j);
 }
```

```
Output: i=0
      j=0 1 2 3 4
       i=1
      j=0 1 2 3 4
```

Break statement(break)

Sometimes it becomes necessary to come out of the loop even before loop condition becomes false then break statement is used. Break statement is used inside loop and switch statements. It cause immediate exit from that loop in whichit appears and it is generally written with condition. It is written with the keywordas **break.** When break statement is encountered loop is terminated and control is transferred to the statement, immediately after loop or situation where we want to jump out of the loop instantly without waiting to get back to conditional state.

When break is encountered inside any loop, control automatically passes to thefirst statement after the loop. This break statement is usually associated with **if**statement.

Example :

```
void main()
{
int            j=0;
for(;j<6;j++)
if(j==4) break;
}
```

```
Output:
0 1 2 3
```

Continue statement (key word continue)

Continue statement is used for continuing next iteration of loop after skipping some statement of loop. When it encountered control automatically passes through the beginning of the loop. It is usually associated with the if statement. It isuseful when we want to continue the program without executing any part of the program.

The difference between break and continue is, when the break encountered loop isterminated and it transfer to the next statement and when continue is encounter control come back to the beginning

position.
In while and do while loop after continue statement control transfer to the testcondition and then loop continue where as in, for loop after continue control transferred to the updating expression and condition is tested.
Example:-
```
void main()
{
int n;
for(n=2; n<=9; n++)
{
if(n==4)  continue;
printf("%d", n);
  }

}
Printf("out of loop");
}
```
Output: 2 3 5 6 7 8 9 out of loop