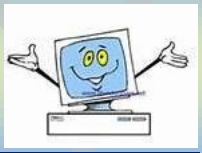
12th class computer science (Session: 2021-22)

Lesson 2

Control statements in c

Things to remember



1. C language provides three types of control statements: branching, looping and jumping.
2. Branching statements are called branching because these statements select one branch and another branch during the running of the program.
3. Loops provide us with a way to repeat statements and also control how often statements have to be repeated.
4. We can use any number of else-if blocks between if and else.
5. Writing one if statement into another if statement is called nested-if statement.
6. Switch_case is like an if-else-if statement.
7. In pre_test loops, the test conditions are checked before the body of the loop.
8. In post test loops, test conditions are tested after the body of the loop.

9. For loop helps us to create loops in which statements can be repeated up to a certain number.

10. 'for' and 'while' loop are examples of pre_test loops in C language while do-while is post test example of loops.

11. In C programming, the jumping statement is used to change the normal flow of a program.

12. The goto statement is used in programming, to jump from the goto statement to the label statement within the same function without any condition test.

13. The break statement is used to encrypt the case in the switch statement.

14. continue is used to release certain parts of the body of the loop.

ਅਭਿਆਸ :

PART -1

Q1. Multiple Choice Questions:-

1) Which of the following statements is called a conditional statement?

a) for b) break c) if d) while

2) switch-case is like ____________________ statement?

a) if else b) if-else-if c) break d) goto

3) Used to end the case in switch statement?

a) continue b) goto c) if d) break

4) Which of the following is an example of a post test loop?

a) for b) while c)do-while d) continue

5) Which of the following is not a jumping statement?

a) while b continue c) goto d) Break

Q:2 Fill in the blanks:-

i) The control condition in the loop is tested before the body of the loop.______(while)

ii) The control condition in the _____ loop is tested after the body of the loop.(do-while)

iii) The ___________ statement is used to skip some statements in the loop.(continue)

iv) _____________ is a multi-way control statement.(switch)

v) Break statements are used to end a case in ____________ statements.(switch)

ਪ੍ਰQ:3 Very Short Answer Questions:

Q:i) What is it called to write one if statement in another if statement?

Answer: Writing one if statement into another if statement is called Nested if.

Q:ii) What statements are used to change the normal flow of programs in C language?

Ans: The C language uses control statements to change the normal flow of programs.

Q:iii) Which statement is most useful for dropping some statements inside the loop?

Ans: Continue statement is most useful for dropping some statements inside the loop.

Q: iv) Which statements provide a way to repeat commands?

Ans: Looping statements provide a way to repeat commands?

Part -2

Q:4 Short Answer Questions (Write the answer in 4-5 lines)

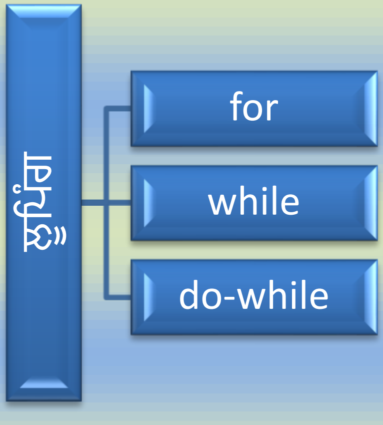
Q:i) Define branching. Write the names of its various control statements.

ਉੱਤਰ: Branching means applying a set of instructions that depends on the outcome of a decision. These statements select one branch or another during the running of the program, hence they are called branching statements. These statements are used for decision making and multi-way selection. These statements are divided into two main sections: -

1. Conditional Control Statement
2. Multi-Way Conditional Control Statement

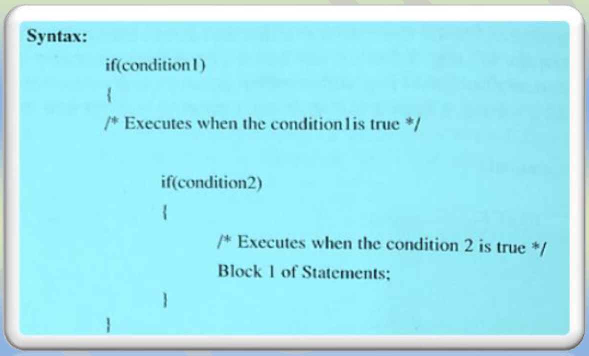
Q:ii) What is looping? Write the names of three different types of looping statements.

Ans: Looping means repeating a set of instructions over and over again. Looping statements are also called iterative statements. Sometimes there may be situations in the program where we need to execute a block of statements multiple times. There are three types of looping statements:



Q: iii) What is a nested-if statement? Write its structure.

Ans: Writing an if statement into another if statement is called a nested if statement. The internal statement only works when the external condition is true. Thus we can say that the block of the inner if statement will work when both the condition of the external if statement and the condition of the internal if statement are true.



Syntax:

```
        if(condition1)
        {
        /* Executes when the condition1is true */

                if(condition2)
                {
                        /* Executes when the condition 2 is true */
                        Block 1 of Statements;
                }
        }
```

Q:iv) What is an if-else statement? Write a program for the if-else statement.

Ans: The if-else statement is used when selecting one of two statements. In the if-else conditional control statement the statements in the if block are executed only when the test condition is true and the statements in the else block execute only when the test condition is false. The syntax for using this statement is as follows:

Program: The following is a program that determines whether a student has "passed" or "failed" according to the marks obtained by the student admitted by the board.

```
#include<stdio.h>


#include<conio.h>
void main()
{
        int marks;
        clrscr();
        printf("Enter Marks:-");
        scanf("%d",&marks);
        if(marks>=33)
        {
                printf("PASS");
        }
        else
        {
                printf("FAIL");
        }
getch();
```

```
OUTPUT:-
Enter Marks:- 34
PASS
Press any key to continue
```

Q:v) What is a while statement? Write its structure.

Ans: Statements are repeated in the while loop as long as the given test condition remains true. This loop is also an example of a pre-test loop. In this loop the test condition is checked first and the statements are repeated later only when the result of the test condition is true. The minimum number of repetitions in this loop is zero because if the condition is initially false then the loop will not run even once. The syntax and program are given below:

```
Syntax:-
while(condition)
{
      Statements;
}
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int i=10;
        clrscr();
        while(i>=1)
        {
                printf("\n%d",i);
                i=i-2;
        }
        getch();
}
```

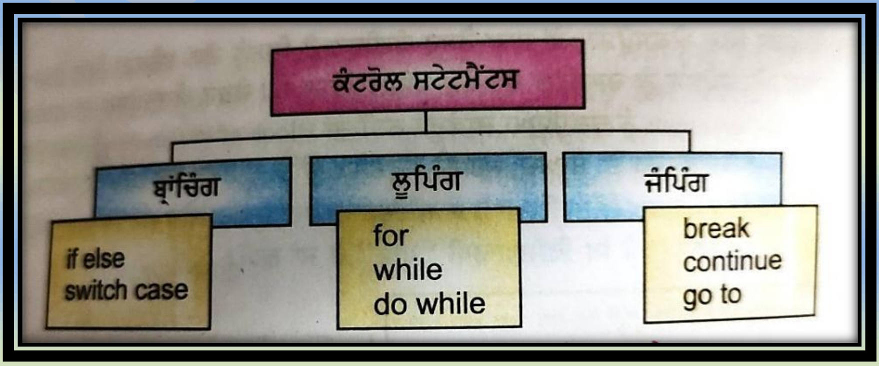
```
OUTPUT:-
10
8
6
4
2
Press any key to continue
```

Answers to Big Questions: - (Write the answer in 10-15 lines)

ਪ੍ਰ:Q1) What are control statements? Write their types?

Ans: The statements that we use to control the flow of the program are called control statements. There are a variety of effective and flexible statements in the C language. With the help of these control statements we can transfer the control point from one place to the required places in the program or even to repeat a statement in the program.. These control statements can be divided into three parts: -

1. Branching control statements (if, if-else, if-else-if, nested-if, switch)
2. Looping Control statement (for, while, do-while)
3. Jumping control Statement (break, continue, goto)



The use of branching statements decides what to do, looping statements determine how often a task is to be performed, and the use of jumping statements to transfer control from one place to another in the program without any condition checked.

Q:2) What is a switch statement? Write a program for switch statement.

Ans: We use switch case statements for multi-way conditional control statements in C programs. This is similar to if else if statements. The switch statement selects one of the many statements. It is based on the integer value or char. In the switch case, a set of statements is applied only if the corresponding case constants are true. When all case statements are false, the control point returns to default.

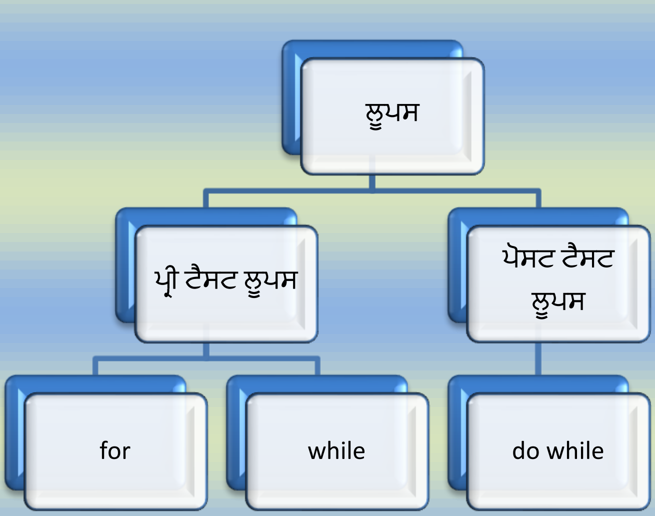
```
#include<stdio.h>
#include<conio.h>
void main()
{
        Int day;
        clrscr();
        printf(“ENTER DAY NUMBERS :- );
        scanf(“%d”,&day);
        switch(day)
        {
                    case 1: printf(“Monday”); break;
                    case2: printf(“Tuesday”); break;
                    case3: printf(“Wednesday”);
        break;
                    case4: printf(“Thursday”); break;
                    case5: printf(“Friday”); break;
```

OUTPUT:-

```
ENTER DAY NUMBERS :- 5
Friday
Press any key to continue
```

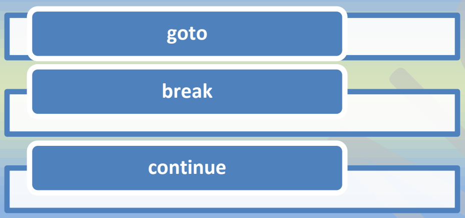
Q3:) What are loops? What are the two different categories of loops?

Ans: Looping means repeating a set of instructions over and over again. Looping statements are also called iterative statements. Sometimes there may be situations in the program where we need to execute the block of statements multiple times. In such cases loops provide us with a way to repeat the statements and they also control how often the statements are to be repeated. All looping statements are divided into two categories as under:



Q:4) What are jumping statements? Describe the types?

Ans: Jumping means shifting control from one point to another. In C programming, the jumping statement is used to change the normal flow of a program. The following jumping statements are used in C language: -



goto

break

continue

1) **goto Statement :-**

The goto statement is used in programming to jump from the goto statement to the label statement within a single function without any condition testing. The use of the goto statement usually depends on the if condition. Syntax:-

```
goto label;
.............
label: statement;
```

2) **break Statement:-**

The break statement terminates the loop or switch statement and transfers the control flow to the next statement immediately after the loop or switch. Syntax:-

```
{
.............
break;
.............
}
```

3) continue Statement:-

Sometimes it is helpful to leave some statements inside the loop. In such cases, the continue statement is used. Syntax:-

```
{
………….
continue;
………….
}
```

પ્ર:5) What is do-while loop? How is it different from the while loop?

Ans::- The do-while is also called a pre-test loop. This is because the statements are applied first and then the condition is tested. It differs from the while loop by the following: -

1. In the do-while loop the statement runs first, the condition is checked later but in the while loop the condition is checked first, the statement runs only after the condition is correct.
2. The do-while loop is guaranteed to run at least once as it operates the body of the loop without testing the condition. But the minimum number of repetitions of statements in the while loop is 0.
3. In the do-while loop, a semicolon with} while (condition); is used. But semicolon with {while (condition) is not used in while loop.

Syntax (do-while)	Syntax (while)

do
{
……………………..
statements;
……………………..
}while(condition);
	while(condition)
	{
	……………………..
	statements;
	……………………..
	}