# CHAPTER 4

# CONTROL STRUCTURES

**After completing this lesson, you will be able to:**

- Explain the use of following decision statements
  If statement
  If else statement
  Else if statement
  Switch statement
- Know the concept of nested if
- Use break statement and exit function
- Explain the use of the following looping structures
  For loop
  While loop
  Do while loop
- Use continue statement
- Know the concept of nested loop

## INTRODUCTION

**Q1. Give a brief information to control structures. Name the types of control structures and briefly explain them.**

**Answer**

**Control Structures**

Control structures are a very important concept in computer programming. Control structures allow programmers to control the flow of program execution

**Types of Control Structures**

Three types of control structures are used in programming which are given below

i. Sequential structure
ii. Conditional structure
iii. Repetition structure

i. **Sequential structure**

Sequential structure refers to execution of instructions one by one in the sequence in which they appear in the program from top to bottom till the last instruction.

ii. **Conditional structure**

Conditional structure also known as decision making structure. Refers to execution of instructions based on a condition. If a condition is met a specific set of instructions are executed otherwise control is transferred to some other part of the program

iii. **Repetition Structure**

Iteration structure also known as loop, refers to execution of same set of instructions several times till a condition is met

## 1.1 DECISION CONTROL STRUCTURES

**Q2. Give a brief introduction to decision making statements and its types**

**Answer**

**Decision making statements**

Decision making structures are control structures that are used in programming to make decisions. They allow programs to execute a specific statement or a set of statements based on one or more conditions.

**Types of Decision making statements**

The decision making statements available in C++ language are as follows

**i.** If statement

**ii.** If else statement

**iii.** Else if statement

**iv.** Switch statement

## 1.1.1 IF STATEMENT

**Q3. What is the use of IF statement? Write down its general form and explain with program**

**Answer**

**If Statement**

The if statement is used to execute a block of statements based on a condition

**General Form**

The following is a general form of if statement

```
If (condition)

{

Block of statements

}
```

**Explanation**

- **i.** The condition is evaluated
- **ii.** If the condition is true, the block of statements within the braces, following the condition is executed
- **iii.** If the condition is false the block of statements is skipped and control is transferred to the next statement after the closing brace
- **iv.** If there is a single statement to be executed then braces is not required

**Program 1**

The following program will read two numbers. If the first number is a positive number it will print the sum and product of the two numbers

```
#include<iostream>

#include<conio.h>
```

```
Void main()
{
Int x,y,sum,prod:
Cout<<"\nEnter first number: ";
Cin>>x;
Cout<<"\nEnter second number: ";
Cin>>y;
If (x>0)
{
        Sum=x+y;
        Prod=x*Y;
        Cout<<"\nSum="<<sum<<end1;
        Cout<<"\nProduct="<<prod;
}
Getch():
}
```

**Output of the Program**

The following is the execution of the program

```
Enter first number : 3
Enter second number : 4
Sum=7
Product=12
```

**Explanation**

The program prompts the user to enter two numbers which are stored in variables x and y. if the first number (x) is greater than zero then the sum and product of the numbers are calculated and stored in variables sum and prod and their values are printed.

### Program2

The following program reads marks and prints the message "You have passed" if marks are greater than or equal to 33

```
#include<iostream>

#include<conio.h>

Void main()

{

Int marks:

Cout<<"\nEnter the marks: ";

Cin>>marks;

If (marks >=33)

    Cout<<"You have passed";

Getch():

}
```

### Output of the Program

The following is the execution of the program

```
Enter the marks: 48

You have passed
```

### Explanation

In this program, there is a single statement to be executed when the condition Is true. Therefore braces are not used.

## 1.1.2 IF ELSE STATEMENT

**Q4. What is the use of if-else statement? Write down its general form and explain with program**

**Answer**

**If-else statement**

The if else statement allows making decision between two courses of action based on a condition

**General Form**

The following is the general form of if else statement

```
If (condition)
{
Block of statements-1     / / if the condition is true
}
Else
{
Block of statements-2   / / if the condition is false
}
```

**Explanation**

1. The condition is evaluated

2. If the result of evaluation is true. The first block of statements 1 is executed, the second block of statements 2 is skipped and then control is transferred to the next statement
3. If the condition evaluates to false, the first block of statements 1 is skipped and the second block of statements 2, following the keyword else is executed
4. If there is a single statement to be executed whether the condition is true or false then braces are not required

**Program1**

The following program reads a number and prints whether it is even or odd numbers

```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int n,r;

Cout<<"\nEnter a number: ";

Cin>>n;

R = n % 2;

If(r==0);

    Cout<<n<<"is even number";

Else

    Cout<<n<< "is odd number";

Getch()

}
```

**Output of the Program**

The following is the execution of the program

```
Enter a number : 15

15 is odd number
```

**Explanation**

When this program is executed it prompts the user to enter a new number. The number is stored in variable n. the modulo operator (%) gives the remainder after division of the entered number by 2 and it is stored in variable r. if the remainder r is equal to 0 then the program will print n is even otherwise it is odd

Braces are not used in if else statement in this program because a single statements is to be executed whether the condition is true or false.

## 4.1.3 ELSE IF STATEMENT

**Q5. What is the se of else if statement? Write down its general form and explain with program**

**Answer**

**Else if statement**

The else if statement is used in situation where a decision is to be made from several alternatives based on various conditions

**General Form**

The following is the general form of else if statement

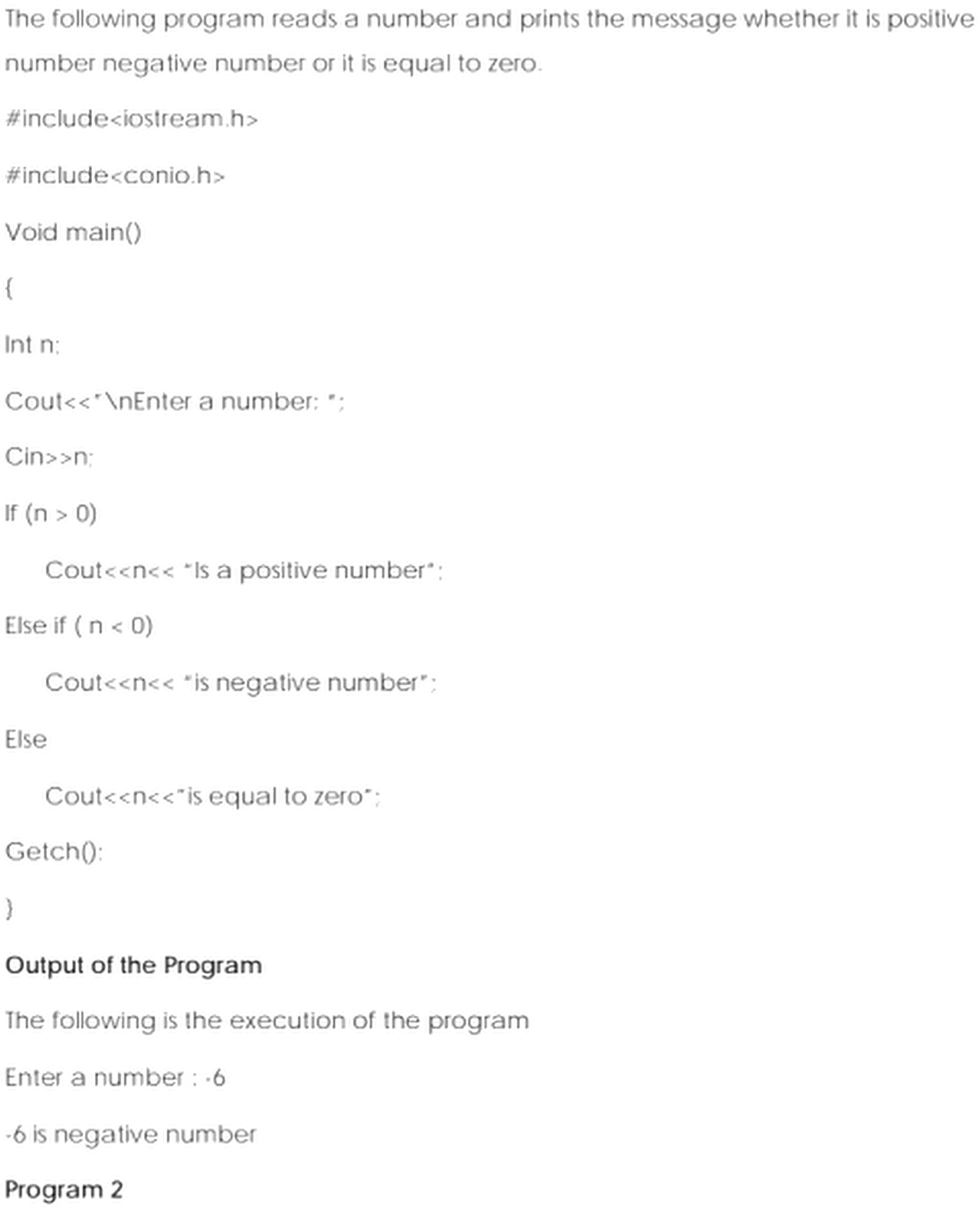
```
If (condition-1)

{

        Block of statements-1

}
```

```
Else if (conditions -2)
{
        Block of statements-2
}



Else
{
       Block of statements-n
}
```

**Explanation**

1. The condition 1 is evaluated
2. If it is true the Block of statements-1 is executed and control is transferred to the next statement
3. If the condition-1 is false then condition-2 is evaluated. If it is true then the Block of statements-2 following condition-2 is executed
4. In this manner, conditions are evaluated one by one. When any condition is true, the block of statements following that condition is executed rest of the code is skipped and control is transferred to the next statement
5. If none of the conditions is true then the last block of statements following the keyword else is executed the else block is optional
6. If a single statement is to be executed instead of a block of statements then braces are not required

**Program1**



The following program reads a number and prints the message whether it is positive number negative number or it is equal to zero.

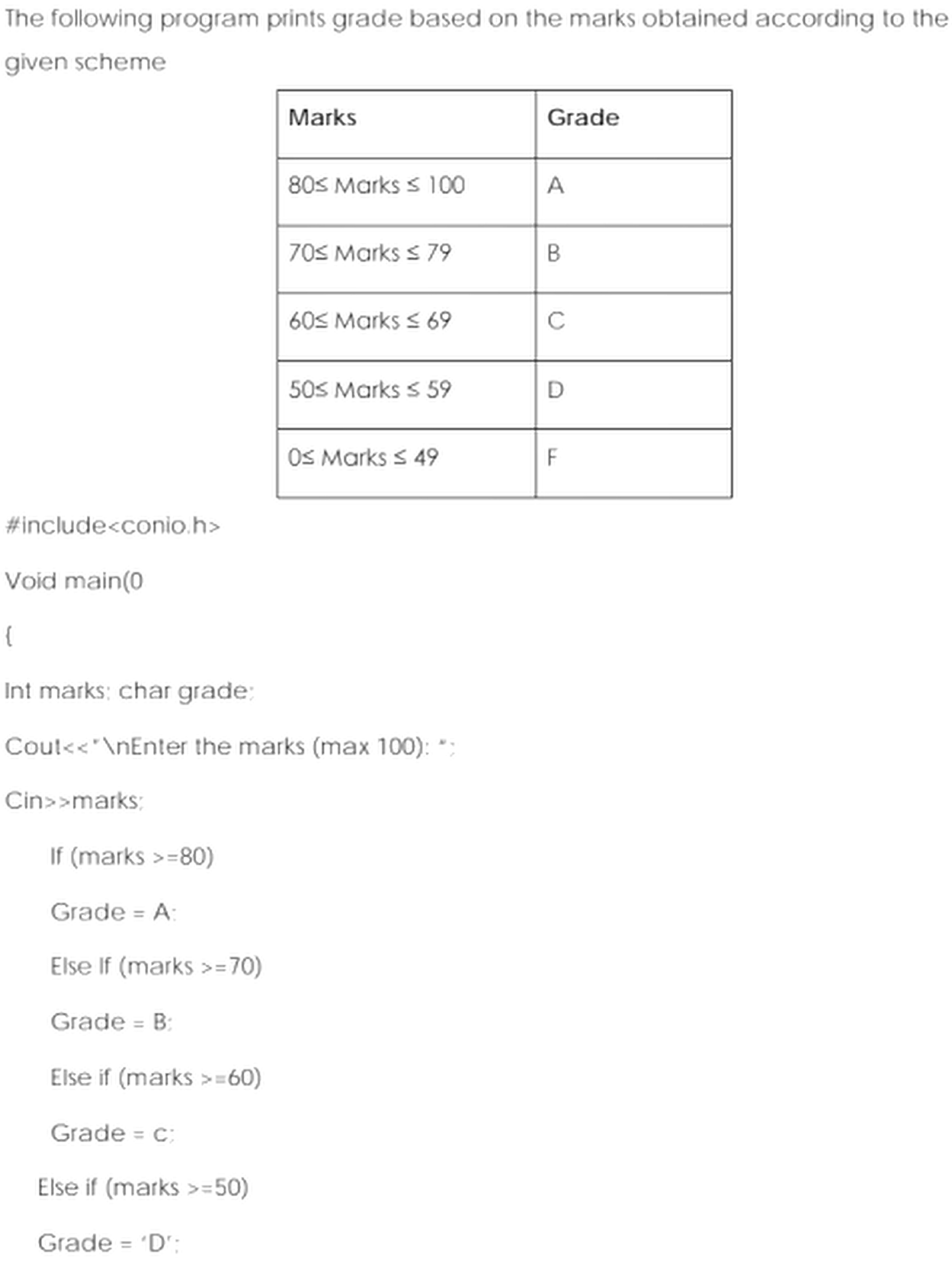
```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int n;

Cout<<"\nEnter a number: ";

Cin>>n;

If (n > 0)

    Cout<<n<< "Is a positive number";

Else if ( n < 0)

    Cout<<n<< "is negative number";

Else

    Cout<<n<<"is equal to zero";

Getch();

}
```

**Output of the Program**

The following is the execution of the program

```
Enter a number : -6

-6 is negative number
```

**Program 2**



The following program prints grade based on the marks obtained according to the given scheme

| Marks | Grade |
|---|---|
| 80≤ Marks ≤ 100 | A |
| 70≤ Marks ≤ 79 | B |
| 60≤ Marks ≤ 69 | C |
| 50≤ Marks ≤ 59 | D |
| 0≤ Marks ≤ 49 | F |

```
#include<conio.h>
Void main(0
{
Int marks; char grade;
Cout<<"\nEnter the marks (max 100): ";
Cin>>marks;
    If (marks >=80)
    Grade = A;
    Else If (marks >=70)
    Grade = B;
    Else if (marks >=60)
    Grade = c;
   Else if (marks >=50)
   Grade = 'D';
```

```
    Else
     Grade = ‘F’;
Cout<< :\n Your grade is” <<grade;
Getch();
}
```

**Output of the Program**

The following is the execution of the program

```
Enter the marks: 74
Your grade is B
```

## 4.1.4 SWITCH STATEMENTS

**Q6. What is the use of switch statement? Write its down its general form and explain with program**

**Answer**

**Switch statement**

The switch statement is a control statement that is used in programming when a single block of statements is to be selected among many choices. It is very similar to else if statement

**General Form**

The following is the general form of switch statement

```
Switch (expression/variable)
{
Case constant-1:  Block of statements
```

```
                              Break:

Case constant-2:  block of statements

                              Break:

Case constant-3  block of statements

                             Break;

.

Default:                block of statements

                            Break;

}
```

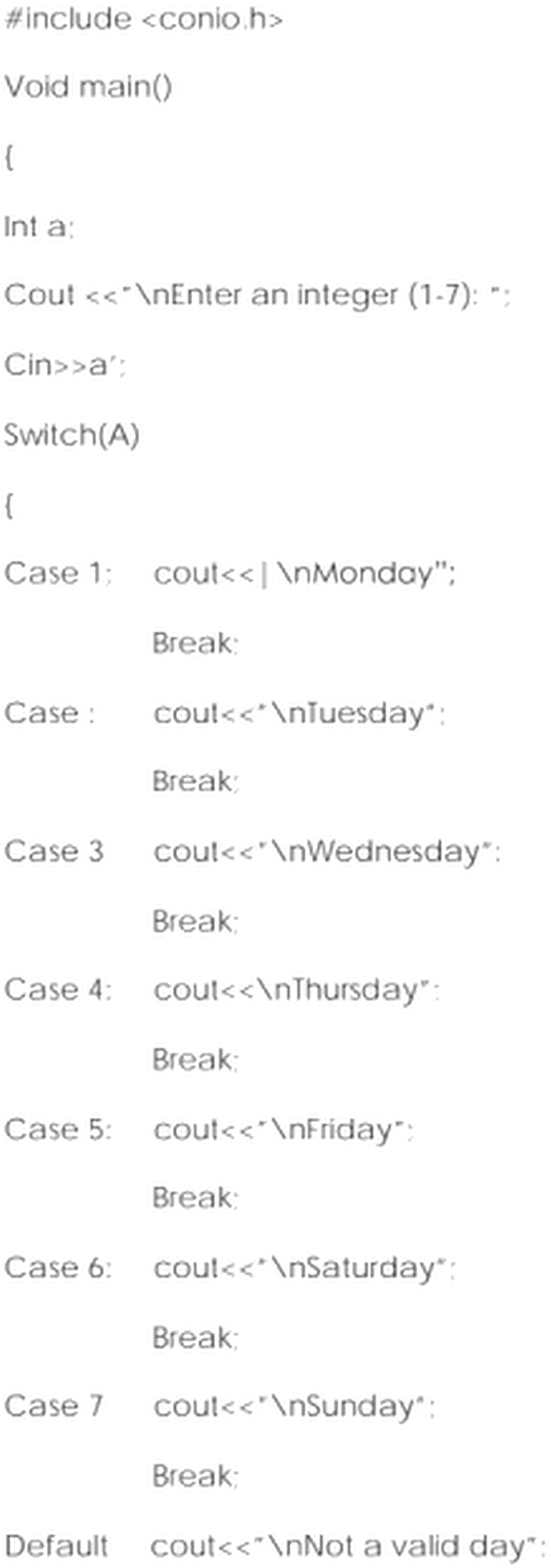
**Explanation**

1. The expression within the brackets is evaluated
2. The result of the expression or the value of variable is compared in sequence with the constant values given after the keyword case
3. If the result matches any constant value then the block of statements following that case is executed and control exits from the body of the switch statement and goes to the first statement following the end of the switch statement
4. If none of the constant value, after the case keyword match with the result of expression or the value of variable then the block of statements following the keyword default is executed. Its use is optional
5. The purpose of break statement is to exit the body of the switch statement

**Program 1**

The following program reads an integer between 1 to 7 that represents a day of week starting from Monday. It prints the name of the day based on the value of day

```
#include <iostream>
```



```
#include <conio.h>

Void main()

{

Int a;

Cout <<"\nEnter an integer (1-7): ";

Cin>>a';

Switch(A)

{

Case 1:    cout<<| \nMonday";

           Break;

Case :     cout<<"\nTuesday";

           Break;

Case 3     cout<<"\nWednesday";

           Break;

Case 4:    cout<<\nThursday";

           Break;

Case 5:    cout<<"\nFriday";

           Break;

Case 6:    cout<<"\nSaturday";

           Break;

Case 7     cout<<"\nSunday";

           Break;

Default    cout<<"\nNot a valid day";
}
Getch();
}
```

**Output of the Program**

The following is the execution of the program

```
Enter an integer 6
Saturday
```

**Program 2**

The following program prompts the user to enter a lower case letter and determines whether it is a vowel or a consonant

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Char ch;
Cout<<"\nEnter a lower case letter: '';
Cin>>ch;
Switch(ch)
{
Case a:
Case e:
Case I:
Case o:
```

```
Case u;

        Cout<<"\nYou entered a vowel:'';

        Break

Default:    cout<<"\nYou entered a consonant: ";

}

Getch():

}
```

**Explanation**

When this program is executed, it prompts the user to enter a lower case letter which is stored in the character variable ch. If the user enters a lower case vowel the message "You entered a vowel" is printed otherwise the default message "You entered a consonant" is printed. The program only checks for the vowels (a, e, i, o, u) using the switch statement. In the absence of statements after the keyword case, the control goes right through one case to the case below and this makes it easy for several values of the switch variable to execute the same code

**Output of the Program**

The following is the execution of the program

```
Enter a lower case letter g

You entered a consonant
```

**Q7. Make a comparison between decision control structures**

**Answer**

**Comparison between Decision Control Structures**

**If statement**

If statement is used when a single block of statements is to be executed if the condition is true. If that condition is false when the block of statements is skipped and the program execution continues from the statement after if block

### If else statement

If else statement is used when one block of statements is to be executed if the condition is true. If that condition is false then the block of statements is skipped and the program execution continues from the statement after if block

### If else statement

If else statement is used when one block of statements is to be executed if the condition is true. If the condition is false then the block of statements following if are skipped and the other block of statements following else is executed. It selects one block of statements from two options

### Else if statement

It selects a block of statements from multiple options. When it is executed, a block of statements is selected for execution based on a condition starting from the first condition. If all the conditions are false then it executes the block of statements after else if it exists otherwise program execution continues from the statement after if block

### Switch statement

Switch statement is a multiple selections statements that is similar to else if statement. It is used to select a block of statements based on the value of a variable or result of an expression. It compares the value of a variable or the result of an expression that is within the parenthesis after the keyword switch, with values specified in each case. If there is a match, the block of statements following that case is executed. If there is no match the default block of statements is executed if it exists otherwise the control is transferred to the statement after switch block.

## 4.1.5 DIFFERENCE BETWEEN ELSE IF AND SWITCH STATEMENTS

Q8. What is the difference between else if and switch statement?

Answer

Difference

| | ELSE IF | SWITCH |
|---|---|---|
| 1 | Which statement will be executed depend upon the output of the expression inside if statement | User decided which statement will be executed |
| 2 | Else if: statement uses multiple decision statements for multiple choices | Switch statements uses single expression for multiple choices |
| 3 | Else if statement test for equality as well as for logical expression | Switch statement test only for equality |
| 4 | Else if statement evaluates integer, character or logical type expressions | Switch statements evaluated only character or integer value |
| 5 | Either if statement will be executed or else statement is executed | Switch statement execute one case after another until a break statement is appeared or the end of switch statement is reached |
| 6 | If the condition inside if statement is false then by default the else statement is executed if created | If the condition inside switch statements does not match with any of cases for that instance the default statements is executed if created. |

## 4.1.6 NESTED IF STATEMENT

Q9. What is meant by nested if statement? Explain it with programs

### Answer

### Nested if Statement

An if statement inside another if statement is known as nested if statement. The C++ language allows to nest if, if else or else if inside another if. If else statement or else if statement

### Programs 1

The following program demonstrates the use of nested if else statement inside another if else statement

```
#include<iostream>

#include<conio.h>

Void main()

{

Int marks;

      Cout<<"\nEnter the marks;

       Cin>>marks;

    If((marks>=0)&&(marks<=100))     //outer if else statement

    If (marks>=33;                   //nested if statement

           Cout>>"You are pass";

    Else

           Cout>>"You are fail";

    Else

           Cout<<"Invalid Data";

 Getch();

}
```

When the above program is executed the user is prompted to enter marks. The first if else statement checks whether the marks entered are in the range of 0 to 100. If it is true then the nested if else statement checks whether the user passed or failed and prints the appropriate message. If the marks entered are out of range then the message "Invalid data" is printed

**Output of the program**

The following is the execution of the program when marks are in the range of 0 to 100

```
Enter the marks: 24

You are failed
```

**Output of the Program**

The following is the execution of the program when marks are in the range of 0 to 100

```
 Enter the marks 113

Invalid data
```

## 4.2 LOOPS

**Q10. Give a brief introduction to loops in C++. States its types**

**Answer**

**Loop**

A loop is a control structure that repeatedly executes a sequence of statements until condition is true. Loops are also called repetition control structures In C++

**Types of Loops in C++**

There are three types of loops in C++ which are follows

- **i.** For loop
- **ii.** While loop

iii. Do while loop

## 1.2.1 FOR LOOP

**Q11. What is the use of FOR loop? Write its down its general form and explain with program**

**Answer**

**FOR loop**

A for loop is used to execute one or more statements for a specific number of times. It is also known as counter loop

**General Form**

The following is the general form of for loop

For (initialization: condition; increment/decrement)

{

Block of statements

}

**Explanation**

1. A variable, known as loop counter or loop variable, is assigned an initial value in the initialization part of the loop. For example, a=1 or b=50
2. The condition which is a relational expression such as a< 5 is evaluated. If the condition is true then the block of statements within the braces is executed
3. After the execution of the statements, control is transferred to the increment/decrement part of the loop. This part consists of an assignment statement such as a= a+7 or a+a-1 that increments or decrements the loop variable

4. If only a single statement is to be executed in for loop then braces are not required.

**Program 1**

The following program prints four times the output of statements that are in the for loop

```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int k;

For (k=1; k<5; k++) // loop executes four times till k is less than 5

{

Cout<< \nI am a student" << end1;

Cout<<"I was born in 2001 "<<end1;

}

Getch();

}
```

**Explanation**

When this program is executed, the loop counter (k) is initialized to 1. The loop condition k<5 is checked. Since it is true , the loop executes and displays the output of two statements that are within the curly brackets the loop counter is incremented by 1. The conditions is checked and the loop statements ar again executed

Each time the loop is executed, the counter k is incremented by 1. The loop continues to execute till k is less than 5. When k becomes 5, it terminates and control is transferred to the next line. In this program k++ is used which is same as k=k+1.

### Output of the program

The following is the execution of the program

```
I am a student

I was born in 2001


I am a student

I was born in 2001


I am a student

I was born in 2001


I am a student

I was born in 2001
```

### Program 2

The following program uses for loop to print all the positive odd numbers that are less than twenty ( 1 3 5 7 9 11 13 15 17 19) on a single line

```
#include<iostream>

#include<conio.h>

Void main()

{

Int k:

For (k=1: k<20: k=k+2)

Cout<<k<<
Getch():
}
```

**Explanation**

The loop variable k is initialized to 1 condition is checked and its value is printed. After printing the value of k control is transferred to the increment part of the loop. The value of k is incremented by 2 and then printed again. The process continues as long k is less than 20. The loop contains a single statements so braces are not used.

## 4.2.2 WHILE LOOP

**Q12. What is the use of WHILE loop? Write its down its general form and explain with program**

**Answer**

**While Loop**

A while loop is a sentinel loop statement. In this loop, the condition is checked at the beginning of the loop. The body of the loop executes as long as the condition remains true. The control can exit a loop in two ways, when the condition becomes false or using break statement

**General Form**

The following is the general form of while loop

```
While (condition)
{
    Block of statement
}
```

**Explanation**

1. The condition which is relational expression such as k<10 is evaluated
2. If the condition evaluates to true, the block of statements with braces is executed
3. After the execution of statements, control is transferred back to the beginning of the loop and the condition is again evaluated. If It is true, then the body of the loop is executed again
4. If the body of the loop consists of a single statement then braces are not required

**Program 1**

The following program prints the sum of all the positive numbers are up to 15 using a while loop

(sum= 1 + 2 + 3 + 4 +....... + 15)

```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int k, sum;

Sum=0;

K=1;            / / initialization of loop variable

While (k<=15)

{

Sum=sum+k;

K=k+1;

                //increment for loop continuation

}
Cout<<"Sum="<<sum;<<End1:

Getch():

}
```

### Explanation

The variable sum is initialized to 0 and the loop variable k to 1. Each time the loop is executed, the value of k is added to sum and it is incremented by 1. This process continues till the condition k<=15 is true. When k becomes 16. The loop terminates.

### Output of the Program

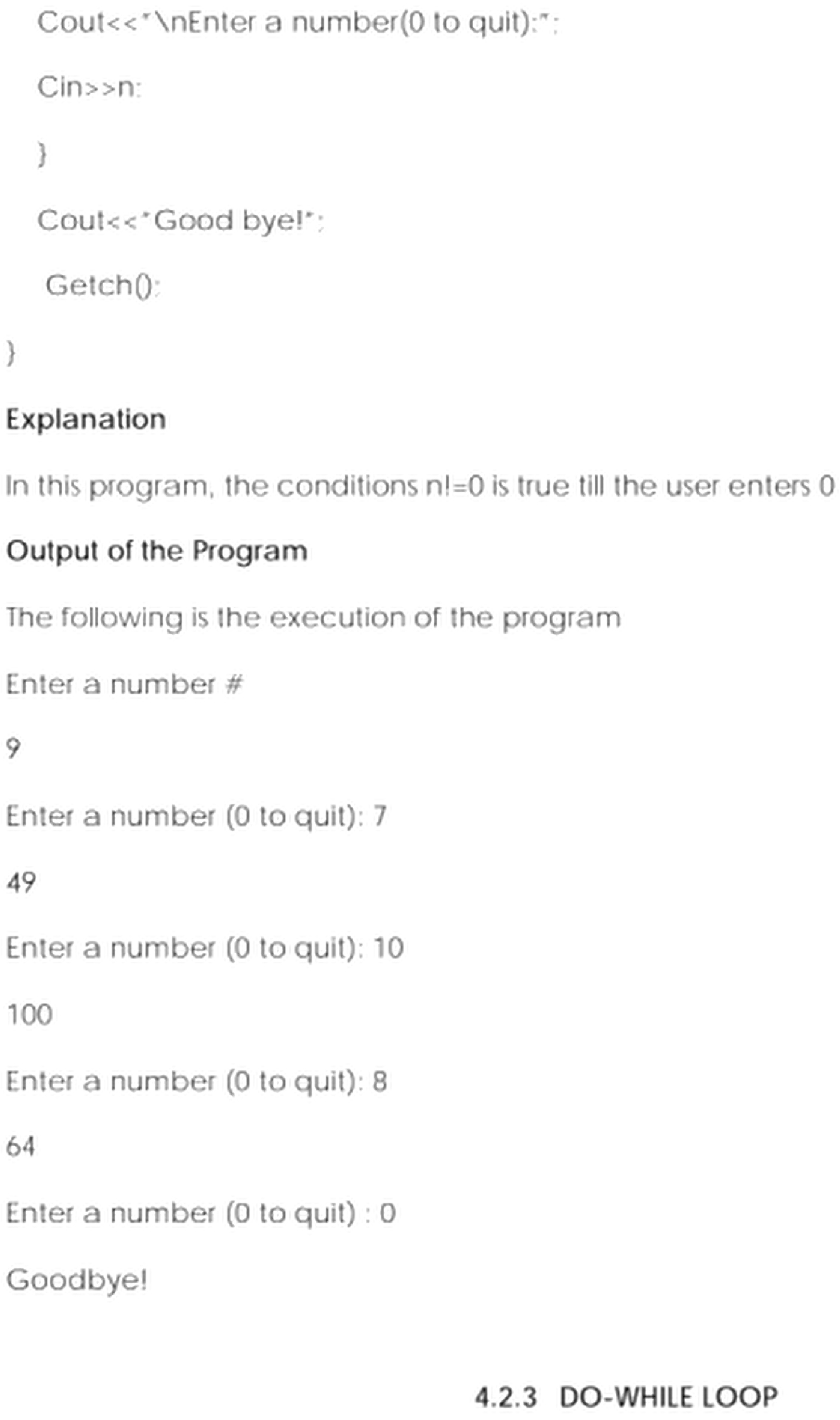
The following is the output of the program

```
Sum=120
```

### Program 2

The following program uses while loop to repeatedly prompt the user to enter a number and prints its square. The loop terminates when the user enters 0 and the program prints the message "Goodbye"

```
#include<iostream>

#include<conio.h>

Void main()

{

Int n;

Cout<<"Enter a number";

Cin>>n'

While (n!=0)  // != is not equal to relational operator

{

   Cout<<n*<<end1;
```



```
   Cout<<"\nEnter a number(0 to quit):";
   Cin>>n:
   }
   Cout<<"Good bye!";
    Getch():
}
```

### Explanation

In this program, the conditions n!=0 is true till the user enters 0

### Output of the Program

The following is the execution of the program

```
Enter a number #
9
Enter a number (0 to quit): 7
49
Enter a number (0 to quit): 10
100
Enter a number (0 to quit): 8
64
Enter a number (0 to quit) : 0
Goodbye!
```

## 4.2.3 DO-WHILE LOOP

### Q13. What is the use of DO-WHILE loop? Write down its general form and explain with program.

Answer

### Do-while Loop

A do while loop is very similar to while loop except that the loop condition is checked at the end of the loop. Therefore, the body of the loop is executed at least once

### General Form

The following is the general form of do while loop

```
Do

{

Block of statements

}

While (condition);
```

### Explanation

1. The block of statements following the keyword do is executed
2. The condition at the end of the loop is evaluated. If the condition is true, control is transferred back to the beginning of the loop
3. The loop is executed once again and then the condition is also checked again. This process continues the condition becomes false
4. When the condition becomes false, control is transferred to the next statement
5. If the body of the loop consists of a single statement then braces are not required

### Program

The following program prompts the user to enter two numbers and prints their product. After printing the product it asks the user whether he/she wants to continue printing product of another set of two numbers. If the user wants to continue he/she enters the character y otherwise the character n

```
#include<iostream.h>

Void main()

Int a, b, prod;

Char ch;

Do

{

Cout<<"\nEnter two numbers";

Cin>>a>>b;

Prod=a*b;

Cout<<"Product="<<prod<<end1;

Cout<<"Do you want to continue? (y/n)";

Cin>>Ch;

}

While (ch!=n);

}
```

**Output of the Program**

The following is the execution of the program

```
Enter two numbers: 3,4

Product=12

Do you want to continue? (y/n): y

Enter two numbers: 7,8
```

```
Product=56

Do you want to continue? (y/n):y

Enter two numbers: 10,12

Product=120

Do you want to continue? (y/n): n
```

## 4.2.4 DIFFERENCE BETWEEN WHILE LOOP AND DO WHILE LOOP

**Q14. What is the difference between While loop and Do While loop?**

**Answer**

**Difference**

| | WHILE LOOP | DO WHILE LOOP |
|---|---|---|
| 1 | While loop is pre tested as the condition is checked in the start | Do while loop is post tested loop as the condition is checked in the end |
| 2 | It is called entry controlled loop | It is called exit controlled loop |
| 3 | The body statements never executes if the condition is false at the beginning | The body statement execute at least once even if the condition |
| 4 | There is no semicolon at the ed of while statement | There is semicolon at the end of do while statement |

## 4.2.5 THE BREAK AND CONTINUE STATEMENTS

**Q15. What is the use of BREAK statement? Explain its uses with the help of program**

### Answer

### The Break Statement

The break statement has two usages

1. It is used to terminate a case in switch statement and program execution continues from the next statement following the switch statement
2. The break statement is also used to terminate a loop when it is encountered inside a loop and program execution continues from the next statement following the loop

### Program

The following program demonstrates the use of break statement inside a for loop

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Int k, count=0;
For (k=1; k<10; k++)
{
Cout<<k<<" ";
Count=count+1;
If (k==4)
Break;
}
Cout<<"\nThe loop executed "<<count<<"times."<<end1;
Getch();
}
```

### Output of the Program

The output of the program will be

```
1  2  3   4
```

The loop is executed 4 times

### Explanation

In this program, when the value of loop variable k becomes 4, the loop terminates and program execution resumes from the next statement following the loop

### Q16. What is the use of CONTINUE statement? Explain its use with the help of diagram

### Answer

### The Continue Statement

The continue statement is used inside a loop. When it is encountered, it transfers control to the beginning of the loop skipping the remaining statements

### Program

The following program demonstrates the use of continue statement. It prints all the positive numbers less than 15 skipping those that are greater than 5 and less han 10 i.e. the numbers 6,7,8 and 9 are not printed

```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int n:

For(n=1;n<15,n++)
```

```
{
      If((n>%)&&(n<10)
              Continue:       // if n is greater than 5 and less than 10 then skip
     Cout<<n<<" ":            // printing the value of n and continue the for loop
```

**Output of the Program**

The output of the program will be

```
1 2 3 4 5 10 11 12 13 14
```

## 4.2.6 EXIT() FUNCTION

**Q17. What is the use of exit function? Explain its uses with the help of example**

**Answer**

**EXIT() Function**

The exit() function is used to terminate a C++ program before its normal termination and exit to the operating system. It requires the standard library header files stdlib.h

**General Form**

The following is the general form of exit function

Exit (value):

Here value is an integer variable. It is known as exit code. The exit code 0 exits a program without any error and exit code 1 indicates that an error must have occurred. It helps the programmer in debugging the program

**Explanation**

Consider the following if else statement

```
if(n>0)
```

```
Cout<<n<<"is a positive number" :
Else
Exit(0):
```

In the above if else statement, the control function exit is used to terminate the execution of if else statement if the value of n is not greater than 0 and return to operating system.

## 4.2.7 NESTED LOOP

**Q18. What is meant by nested loop? Explain its use with the help of program**

**Answer**

**Nested loop**

A loop inside another loop is known as nested loop. The C++ language allows to nest a for, while or do while loop inside another for, while or do while loop.

**Program**

The following program demonstrates the use of nested for loop inside another loop

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Int I, j:
For(i=1,i<5;i++)
{
Cout<<"\n":
```

```
For (j=1;j<11;j++)       //transfers printing to next line

            Cout<<"*"    // inner loop

}

Getch();

}
```

### Explanation

In this program, I Is the outer loop variable and j is the nested loop variables. When this program is executed, the outer loop will execute till the value of I is less than 5 which means it will execute 4 times. The variable I will be initialized to 1 and the statement cout<<"\n"; will transfer the control to next line. The nested loop will execute and print 10 asterisks (*) on a single line as shown below

```
                                   **********
```

The variable I will be incremented by 1 and the nested loop will again print another line of 10 asterisks. This process will continue till I is less than 5. Hence, this program will pint 4 lines of 10 asterisks.

### Output of the Program

The following is the output of the program

```
*********

*********

*********

*********
```

### Program

The following program demonstrates the use of nested for loop inside while loop for printing the following pattern

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Int n,k;
N=1;
While (n<=5)        // outer loop
{
Cout<<"\n";         // transfers printing to next line
        For (k=1; k<=n;k++)    // inner loop
              Cout<<k<<" ";
              N++;
}
Getch();
}
```

**Explanation**

In this program n is the outer loop variable and k is the loop variable for nested for loop. When the program execution begins n is initialized to 1. The cout<<"\n; statement instructs to start printing at the beginning of next line. The nested loop executes once since the value of n is equal to 1 and prints 1 at the beginning of the lne. The value of n is incremented by 1 and becomes 2. The nested for loop now executes twice and prints the numbers 1 and 2 on the same line with space in between. The process continues as long the value of n is less than 6

## KEY POINTS

- Decision making structures are control structures that are used in programming to make decisions. They allow programs to execute a specific statement or a set of statements based on one or more conditions
- The if statement Is used to execute a block of statements based on a condition
- The if else statement allows making decision between two courses of actions based on a condition
- The switch statement is a control statement that is used in programming when a block of statement is to be selected among many choices
- An if statement inside another if statement is known as nested if statement
- Loop is a control structure that repeatedly executes a sequences of statements until a condition is reached
- A for lop is used to execute one or more statements for a specific number of times
- A while loop is used when the number of times the loop will execute may not be known in advance
- The do while loop is very similar to while loop, except that the loop condition is checked at the end of the loop. Therefore the body of the loop is executed at least once