CHAPTER 3

OBJECT ORIENTED PROGRAMMING

IN C++

After completing this lesson, you will be able to:

- Define program
- Define header file and reserved words
- Describe the structure of C++ program
- Know the use of statement terminator
- Explain the purpose of comments and their syntax
- Explain the difference between constant and variable
- Explain the rules for specifying variable names
- Know data types used in C++
- Define constant qualifier -const
- Explain the process of declaring and initializing variables
- Use type casting
- Explain the use of cout statement
- Explain the use of cin statement
- Define getch(), gets() and puts() functions
- Define escape sequence
- Use of escape sequence in programs
- Use I/O handling functions
- Use manipulators endl and setw
- Define operators and know their use in programs
- Use unary binary and ternary operators in programs
- Define expression
- Define and explain the order of precedence of operators
- Define and explain compound expression

1.1 INTRODUCTION

Q1. Give a brief introduction to C++

Answer

C++

C++ is a general purpose programming language that supports various computer programming models such as Object Oriented Programming (OOP) and generic programming. Bjrance Stroustrup created it in early in 1980s at Bell Laboratories

Purpose of C++

Main purpose of C++ was to make writing good programs easier and more pleasant for the individual programmer

Use of C++

C++ supports modern programming techniques. It is commonly used for developing high performance commercial software, games and graphic related programs.

1.1.1 COMPUTER PROGRAM

Q2. What is meant by a computer program? Explain.

Answer

Computer Program

A computer program is a set of instructions that performs a specific task when executed by a computer. It tells the computer what to do.

Explanation

Everything a computer does is controlled by computer program

Examples

Microsoft Word is a program that allows computer users to create documents. Skype is a program used to make cells free of charge to other people who are on Skype.

Q3. Define development of program, machine language, source code and object code

Answer

Development of Programs

Generally programs are written in English oriented high level languages such as Visual Basic, Pascal, Java C++ etc

Machine Language

A computer can only understand instruction in machine language, which consists of 0s and 1s. Therefore, a program written in a high level language must be translated into machine language before execution. This task is achieved by software known as compiler

Source Code & Object Code

A program written in high language is called source code and its equivalent program in machine language is called object code

1.1.2 HEADER FILE AND RESERVED WORDS

Q4. What are header files? Give some examples

Answer

Header Files

The C++ contains many header files. Header files contain information that is required by the program in which these are used. It has .h extension.

Examples of Header files

Some examples of header files are iostream.h conio.h and math.h. These files are included in the standard library of C++ compiler.

Q5. What are pre-processor directives? State its syntax & give some examples

Answer

Pre-processor directive

Pre-processor directive is used to include a header file at the beginning of program.

Pre-processor directive is not a normal program instruction to be executed by the CPU.

It is a code for the compiler to include a header file

Syntax of Pre-processor Directive

The syntax of pre-processor directive to include a header file in a program is

```
# include <name of header files>
```

It begins with a #sign followed by the world include and then name of header file is written angled brackets < >

Example 1

To include the header file iostream.h in a program, the code is

```
#include <iostream.h>
```

The effect of this line is to copy all the contents of iostream.h header file into the program and make them available for use. Almost all the C++ programs perform input/output operations. Therefore, generally this header file is included in the programs.

Example 2

The following is another example for including **math.h** header file in a program

```
#include <math.h>
```

The math.h header file contains code for performing mathematical functions such as finding the square root of a number.

Q6. Define reserved words. Make a list of reserved words

Answer

Reserved Words

Reserved words are special words which are reserved by a programming language for specific purpose in program. These cannot be used as a variable names. All the reserved words are written in lower case letters. There are about 80 reserved words in C++ but this may vary depending on the version being used.

Example

Some examples of reserved words of C++ are **if, void, break, while, case** and **char**

List of Reserved Words

The reserved words of C++ are

And	And_eq	Asm	Auto	Bitand	Bitor	Bool	Break
Case	Catch	Char	Class	Const	Const_cast	Continue	Default
Delete	Do	Double	Dynamic_cast	Else	Enum	Explicit	Export
Extern	False	Float	For	Friend	Goto	If	Inline
Int	Long	Mutable	Namespace	New	Not	Not_eq	Operator
Or	Or_eq	Private	Protected	Public	Register	Reinterpret_cast	Return
Short	Signed	Sizeof	Static	Static_cast	Struct	Switch	Template

This	Thro w	True	Try	Typedef	Typeid	Typename	Union
Using	Virtu al	Void	Volatile	Unsigne d	Wchar_t	While	Xor_eq
Xor							

1.1.3 STRUCTURE OF C++ PROGRAM

Q7. Describe the structure of a C++ program

Answer

Structure

A C++ program has the following structure

Pre-processor directives

Void main()

{

Body of main () function

}

Explanation

A C++ program starts with **pre-processor directives,** followed by the line void main() function and then the body of the main() function is written within curly brackets { and }

Body of main() function

Body of main() function consists of executable statements. These statements perform a specific task when the program is executed by the CPU. There is no restriction on the number of statement that can be written in the body of main() function

Program

To understand the structure of program, consider the following program

```
#include <iostream.h>
Void main()
{
Cout<< "Information Technology";
}
```

Explanation

#include<iostream.h>

This is the first line of the program and it is a pre-processor directive. This program prints a message on the screen. Therefore, iostream.h header file is included which contains the code for performing input/output operations

Void main()

This line identifies the main function of the C++ program. All C++ programs must have main() function. If a program consists of more than one function, the location of main() does not matter. In C++ program the main() function is always executed first

Cout << Information Technology

This is a C++ statement. The meaning of this statement is to print the message Information Technology on the screen. This statement is written within the curly brackets and is called body of the main () function.

1.1.4 STATEMENT TERMINATOR (;)

Q8. Give a brief introduction to statement terminator.

Answer

Statement Terminator

In C++, semicolon is a statement terminator. It marks the end of a statement. All the C++ statement must end with a semicolon. The following statement was used in the following code notice that it ends with a semicolon

```
Cout<<"Information Technology";
```

If ";" is missing the compiler will give syntax error number and also the message that ";" is missing. The error number and message may vary depending on the compiler used.

1.1.5 COMMENTS IN C++ PROGRAM

Q9. State the purpose of comments in C++ program? How many types of comments are used in C++? Explain

Answer

Comments

All the programming languages allow comments in programs. Comments are explanatory statements that help the reader in understanding source code. Comments can be entered at any location in the program. Comments are ignored during program execution. Which means they are not executable statements.

Types of Comments

There are two types of comments in C++. These are given below

i. Single line Comments
ii. Multiple line Comments

i. The Single Line Comments (//)

The single line comments start with // (double slash) and continue until the end of the line

Program

The following program demonstrates the use of single line comments

```
/ /  this is a very simple C++ program.
# include <iostream.h>
Void main ()
{
Cout << "Information Technology";
/ / it prints a message on the screen.
}
```

ii. **The Multiple Line Comments (/* and */)**

This type of comments is used for entering multiple line comments in a program. The /* is used at the beginning of comments and */ ends it Program

The following program demonstrates the use of multiple line comments

```
/* this is my first C++ program
It demonstrates the
Structure of C++ program
*/
# include <iostream.h>
Void main ()
Cout <<"Information Technology".
}
```

The /* and */ type comments can also be used to enter comments on a single line as shown below

```
/* this is my first C++ program */
```

1.2 C++ CONSTANTS AND VARIABLES

Q10. State the purpose of constants and variables in programs

Answer

Constants and variables are used in programs to perform calculation of various types. Therefore, it is important to understand how they are used in computer programs.

1.2.1 CONSTANTS AND VARIABLES

Q11. Define constants and its types. Give examples

Answer

Constant

In computer programming a constant is a value that does not change during execution of program

Types of Constants

A constant can be a number, a character or a character string. A character string is a sequence of any number of characters.

Q12. Define variable

Answer

Example of Constants

Some examples of constants are 42, 7.25, 's' and 'computer' respectively. In C++ a single character constant is written single quotes and a string constant within double quotes.

Q12. Define variable

Answer

Variable

A variable is a name of memory location where data is stored. Variables are used in computer programs to store values of different data types. The data stored in a variable may change during program execution.

1.2.2 RULES FOR SPECIFYING VARIABLE NAMES

Q12. Write down the rules for naming a variable

Answer

The following are the rules for naming a variable

1. The first character of a variable name must be alphabet or underscore
2. The characters allowed in a variable name are
   - i. Underscore (_)
   - ii. Digits (0 to 9)
   - iii. Upper case letters (A to Z)
   - iv. Lower case letters (a to z)

Uppercase Letter

An upper case letter is considered different from a lower case letter

Example

The variable **SUM** is different from Sum or sum

Underscore

The underscore (_) is generally used to improve readability

Example

The variable overtime may also be written as **over-time**

3. Special symbols such as $, @, %, # are not allowed
4. Blank space or comma is not allowed
5. Reserved words of C++ are not allowed to be used as a variable name

1.2.3 C++ DATA TYPRS

Q14. Explain the data types in C++

Answer

Data Types

Data types are declarations of variables for storing various types of data. Data types have different storage capacities. In C++ programs data types of a variable must be defined before assigning it a value

Data types in C++

The data types used in C++ programming are as follows

i. Integer
ii. Floating point
iii. Double precision
iv. Character

1. **Integer**

   It is a data type that is used to define numeric variables to store whole numbers such as -3, 0, 367, +2081etc. Integers represent values that are counted such as number of students in a class. The short form of integer is int Numbers that have fractional part such as 3.84 cannot be stored in an integer variable

Types of Integer

The following table shows the integer types, the number of bytes it takes in memory to store the value and the range of numbers it can store.

Floating type	Number of Bytes	Range of Numbers
Int	4 bytes	-2147482648 to 21474483647
Unsigned int	4 bytes	0 to 4294967295
Short int	2 bytes	-32768 to 32767
Unsigned short int	2 bytes	0 to 655535
Long int	4 bytes	-2147483648 to 2147483647
Unsigned long int	4 bytes	0 to 4294967295

2. **Floating point**

   It is a data type that is used to define variables that can store numbers that have fractional part such as 3.75, 2.1, 388.80 etc. these numbers are also known as real numbers. The short form of floating point is float

   Types of Floating Point

   The following table shows the floating point types. The number of bytes it takes in memory to store the value and the range of real numbers it can store.

   The float type variable might occupy different numbers of bytes and their range might also different depending on the computer and the compiler being used

3. **Character**

   It is a data type that is used to define variables that can store only a single character. One byte memory is set aside in memory to store a single character. The short form of character is char. A variable of type char can store lower case letter an upper case letter, a digit or a special character

   Examples

Some examples of characters that can be stored in a variable of type char are 'a', '+' '%' and '5'. Note that characters are written within single quotation marks (' ')

3.2.4 THE CONSTIANT QUALIFIER (const)

Q15. Give a brief introduction to CONST qualifier in C++

Answer

Constant Qualifier

In C++ programming language, const defines a variable whose value cannot be changed throughout the program. When the const qualifier is used with a variable, it no longer remains a variable because its value will not be changed. A variable defined with the const qualifier must be assigned some value

Syntax

Const has the following syntax

Const data_type variable = constant;

Examples

Const int AGE= 34;

Const float LENGTH = 7.5

The variables of type const are generally written in upper case letters.

3.2.5 VARIABLE DECLARATION AND INITILIZATION

Q16. Describe the variable declaration and variable initialization with examples

Answer

Variable Declaration

In C++ all the variables that are going to be used in a program must be declared before use. Declaring a variable means specifying the data types of a variable. It allows the compiler to decide how many bytes should be set aside in memory for storage of value that is going to be assigned to the variable in the program

Examples of variable Declarations

The following are some examples of declaring variables

Int x, y, z:

Float length, breadth, sum:

Char ch:

Explanation of Examples

Here the variables x,y and z are declared as type of Int, length and breadth as of type float and ch as of type char.

Initialization of variable

A variable may be initialized at the beginning of a program when it is declared. Initialization a variable means assigning it an initial value.

Explanation of Examples

In the first declaration statement, the variable x is initialized to integer constant 4 and y to 5. The second statement initialized length to 12.5 breadth to 15.25 and sum to 0.0. The last statement initialized the character variable ch to a

3.2.6 TYPE CASTING

Q17. What is meant by type casting? Explain its types with examples

Answer

Type Casting

Type casting is used in C++ to convert data type from one types to another

Types of type casting

There are two types of type casting

i. Implicit type casting
ii. Explicit type casting

1. **Implicit Type Casting**

   Implicit type casting automatically converts a data type to another

Example

This is explained by the following example

Suppose variable q is declared as of type float and the following calculation is to be performed

q = 15/6

Explanation

When this integer division is performed, the result will also become an integer value 2 which will be implicitly converted to floating point value 2.0 and assigned to the variable 1

If one or both of the integer constants are converted to floating point constant (14.0 or 6.0) this will perform division using floating point mathematics. In this case, the result produced will be 2.5 and it will be assigned to q

2. **Explicit Type Casting**

   In explicit type casting a special operator is used to convert one data type into another

General form

The general form for conversion is (type) expression

Here expression can be an arithmetic expression or a variable

Example

This is explained by the following example

Suppose a and b are variables of type int and q is of type float. The integer value 15 is stored in a and 6 in b and the following division is to be performed

Q = a/b;

Explanation

When this division is performed, integer math will be used and the result produced will be 2, which will be assigned to the variable q

To obtain correct result, type casting should be used to convert at least one of the operands to type float as shown below

Q= (float)a/b;

Now first the value stored in a will be converted to type float and then the division will be performed. The floating-point math will be used and the correct result 2.5 will be produced which will be assigned to q

Similarly, the int type can also be used to convert a floating point value stored in a floating point variable into integer type by truncating the fractional part of the number.

3.3 INPUT/OUTPUT HANDLING

Q18. Describe input/output handling and functions for handling I/O in C++. Also define streams

Answer

Input/Output Handling

In computer programming, providing data into a program from outside source is also known as input and output means to display some data on screen or save in a file on a storage device

Functions for Handling I/O

The functions getch() gets() and puts() are also used for handling input/output operations

Streams

In C++ input/output is performed by using streams. A stream is a sequence of data that flows in and out of the program. The cin and cout are the standard statements that use the **iostream.h** header file for performing I/O operations. Therefore, it must be included at the beginning of program.

1.3.1 THE cout STATEMENT

Q19. Describe cout statements in C++

Answer

Cout Statement

The cout statement is used to output text or values on the screen

Syntax

The following is the syntax of cout statement

Cout<<character string/variable;

Explanation

The keyword cout is used to with insertion operator on its right side which is two less than signs (<<) followed by a character string or a variable. The insertion operator displays of the character string or the value stored in the variable on the screen. It directs the

output to the standard output device which is monitor. Note that the statement ends with semicolon (;). The insertion operator may be used more than once in a single statement

Program- Demonstrates the use of cout statement

The following program demonstrates the use of cout statement

```
#include <iostream.h> / / this header file is used for I/O functions
Void main()
{
Int a:
A=12:
Coul<<"the value stored in a is":
Coul<<a,
}
```

Output of the Program

The value stored in a is 12

Explanation

The first output statement will print the text and the second will print the value stored in variable a. the output of the two cout statements will appear on the same line.

1.3.2 THE cin STATEMENT

Q20. State the use of cin statement in C++.

Answer

Cin Statement

The cin statement is used to input data from the keyboard and assign it to one or more variables

Syntax

The following is the syntax of cin statement

Cin>>variable;

Explanation

The keyword cin is used with the extraction operator on the right side which is two greater than signs (>>) followed by a variable. When this input statement is executed, it causes the program to wait for the user to input data. The data entered by the user is assigned to the variable

Program

The following program demonstrates the use of cin statement

```
#include <iostream.h>

Void main()

{

Int n,

Cout<<'Enter an integer:";

Cin>>n

Cout<<"the number you types is" <<n;

}
```

Explanation

When this program is executed, the first cout statement prompts the user to enter an integer. The cin statement causes the types number to be assigned to variable n. the last statement prints the number stored in n on the screen.

Output of the Program

The execution of program is shown below

Enter an integer: 7

The number you types is 7

Input more than One Data Values

More than one extraction operator can be used in a single cin statement to input more than one data values as shown below

Cin>>a>>b;

The values for variables a and b should be input with space between them

1.3.3 THE getch(), getche(), gets() AND puts() FUNCTIONS

Q21. Explain the functions, which are used for handling I/O operations.

Answer

Handling I/O Operations

These three functions are also used for handling I/O operations

i. Getch() function
ii. Getche() function
iii. Gets() and puts() Functions

1. **The getch() Function**

   In some situations in programming it is required to read a single character the instant it is types without waiting for Enter key to be pressed.

Example

In a game we might want an object to move each time we press one of the arrow keys. It would be awkward to press the Enter key each time we pressed an arrow key

Explanation

The getch() function is used for this purpose. The get means it gets something from an input device and ch means it gets a character. This function uses the conio.h header file. Therefore, it must be included in the program.

Program- Demonstrates the use of getch() function

The following program demonstrates the use of getch() function to read a character from the keyboard and displays it on the screen

```
#include<iostream.h>
#include<conio.h>
{
Char ch:
Cout <<'Enter a character':
Ch=getch():
Cout<<'the character you types is:",,ch:
Getch ():
)
```

The execution of the program is shown below with the input character a

Output of the program

Enter a character

The character you types is a

Explanation

Note that when this functions is used, the input character a is read and stored in variable ch but it is not displayed on the screen

2. **The getche() Function**

If the user wants the typed character to be displayed on the screen then another similar function getche() is to be used. In this function, the letter e is added which means to echo or display the input character on the screen

Explanation

When executing a program, some C++ compilers display the output for a very short time and immediately returns to the editing window. The user is not able to see the program output. Therefore, very often getch() functions is used at the end of the program so that the user is able to see the program output and has to enter any character to return to the editing window. Since the character entered is not used in program, there is no need to store in a variable. This is shown below

Program

```
#include<iostream.h>
#include<conio.h>
{
Char ch:
Cout<<"Enter a character:":
Ch=getche():
Cout<<the character you types is<<ch;
Getch():
}
```

The execution of the program is shown below with the input character 'h'

Output of the Program

Enter a character:

The character you types is h

Explanation

Note that, when this function is used the input character 'h' is read and stored in variable ch and displayed on the screen.

3. **The gets() and puts() Function**

These functions are used to handle input/output of character strings. The gets() function is used string from the keyboard and the puts() is used to display it on the screen. In C++ strings are handled gets() and puts() defined in <cstdio> header file

Program-demonstration of gets() function

```
#include <iostream>
#include <rstdio>
Using namespace std;
Int main()
{
Char str [100];
Cout << Enter a string" :
Gets (str);
Cout << "you entered"<< str;
Return 0;
}
```

Output of the Program

```
Happy New Year
Happy Birthday
```

Program- demonstration of puts() function

```
#include <cstdio>
Int main()
{
Char str1[] = 'Happy New Year';
Char str2[] = 'Happy New Year';
Puts(str1)
/* Printed on new line since '/n' is added */
Puts (str2);
Return 0;
}
```

Output of the Program

```
Happy New Year
Happy Birthday
```

3.3.4 THE ESCAPE SEQUENCE

Q22. What is meant by escape sequences? Explain its structure and uses.

Answer

Escape Sequences

Escape Sequences are special characters used to control the output look on output devices. These characters are not printed. These are used inside the output statement

Structure of Escape Sequences

An escape sequences begins with a backslash (/) followed by a code character. These are called escape sequences because the backslash causes an escape from the normal way characters are interpreted in C++ programming language

Uses of Escape Sequence

Escape sequences are used for special purposes in programming such as to begin printing on the next line to issue a tab to print special characters etc.

3.3.5 COMMONLY USED ESCAPE SEQUENCES

Q23. Which escape sequences are commonly used in C++ programs?

Answer

Commonly Used Escape Sequence

The commonly used escape sequence are \a \b \n \r \\ \' and \\"

Explanation

Suppose we want to print the following message on the screen

Cout<<"There are many versions of";

```
Cout<<"Windows operating system";
```

When the above two statements are executed, the output will appear on a single line as shown below

There are many versions of Windows operating system

If it is desired to display the output in two lines then \n escape sequence can be used in various ways to move cursor to the beginning of next line

```
Cout<<"\nThere are many versions of";
Cout<<"\nWindows operating system";
```

The following two statements can also obtain the same output.

```
Cout<<"Windows operating system";
```

A single output statement can also be used

```
Cout<<"\nThere are many versions of \nWindows operating system";
```

Similarly, the \t escape sequence can also be used to tab over eight characters as shown in the following statement

```
Cout<<"C++\tis\ta\thigh\tlevel\tlanguage";
```

The output of this statement will be

```
C++     is  a   high    level   language
```

List of Escape Sequence

A list of commonly used escape sequences is given in the following table with their meanings

Escape sequence	Purpose	Escape sequence	Purpose
\a	Produces alert (beep) sound	\t	Moves cursor to the next horizontal tabular positions

\b	Moves cursor backward by one position	\\	Produces a backslash
\n	Moves cursor to the beginning of next line	\"	Produces a double quote
\r	Moves cursor to the beginning of current line	\'	Produces a single quote

The following statement uses the \" escape sequences to print the word 'Pakistan' in double quotation marks

Cout<<"this will print the word \"Pakistan\" in double quotation marks":

The escape sequence \ and \\ are also used in the same way to print a single quote or a backslash.

3.3.6 PROGRAMMING WITH I/O HANDLING FUNCTIONS

Q24. Write a program reads that three integers and prints their sum and product

Answer

Product

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Int x,y,z,sum,prod;
Cout<<"\nEnter first number:";
```

```
Cin>>x;
Cout<<"\nEnter third number";
Cin>>z;
Sum=x+y+z;
Prod=x*y*z;
Cout<<"\nsum="<<sum;
Cout<<\nProduct="<<prod;
Getch();
}
```

Output of the Program

The execution of the program is shown below

Enter first number : 3

Enter second number : 4

Enter third number : 5

Sum =12

Product= 60

Explanation

When this program is executed, it prompts the user to enter three numbers. The user enter the numbers, 3,4 and 5 separated by space and presses the Enter key. The program calculated the sum and product and displays on the screen.

Q25. Write a program that reads the base and height of a triangle and prints area using floating point values

Answer

Program

```
#include<iostream.h>
#include<conio.h>
Void main()
{
Float base, height, area;
Cout<<\nEnter the base";
Cin>>height;
Area=(base*height)/2;
Cout<<"\nThe area of triangle is"<< area;
Getch();
}
```

Explanation

In this program two separate input statements are used for reading the values for variable base and height

Output of the Program

The execution of program is shown is below

```
Enter the base 4:5
Enter the height 6:2
The area of triangle is 13.95
```

3.3.7 THE MANIPULATORS endl AND setw

Q26. What is meant by manipulators in C++? Explain endl and stw with programs

Answer

Manipulator

A manipulator is a command in C++ that is used for formatted output. It modifies the output in various ways. There are many manipulators available in C++. The most commonly used manipulators are endl and setw. The iomanip.h header file should be included when manipulators are used in a program. Only the endl manipulator can be used without using ionmanip.h file.

The endl Manipulator

The endl manipulator has the same function as the \n escape sequence. It causes a linefeed in the cout statement so that the subsequent text is displated on the next line

Program- Demonstrates the use of endl manipulator

The following program demonstrates the use of endl manipulator

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
Void main()
Cout<<"\nI am a student"<<end;
Cout<<"I was born in 2001";
Getch();
}
```

Output of the Program

The output of the program will be

I am a student

I was born in 2001

A single cout statement can also be used as shown below to obtain the same output

Cout<< "I am a student"<<end1<<"I was born in 2001":

3. **The setw Manipulator**

   The setw manipulator is used in output statement to set the minimum field width

General Form

It has the general form

Setw(n)

Here n is an integer value that causes the number or text that follows to be printed within a field width of n characters. The number or text is right, justified within the set field width. It is commonly used in program to align numbers or text or output.

Program- Demonstrates the use of setw manipulator

The following program demonstrates the use of setw manipulator in a program

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
Void main()
{
Int price1=8540, price2=325, price3=27800:
Cout<< "Product"      "<<setw(10) <<"Price"<<end1:
Cout<<"Hard disk      "<<setw(10) <<price<<<End1:
Cout<<"Mouse          "<<setw(10) <<price2<<end1:
Cout<< computer       "<<setw(10) <<price3<<End1:
Getch():
```

}

This program will print the values following the setw manipulator within a field width of 10 characters and they will be right justified

Output of the Program

The output of the program is given below

Product Price

Hard disk 8540

Mouse 325

Computer 27800

Program- Using A single Cout Statement

The same program can also be written using a single cout statement as shown in the next program

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
Void main()
Int price1=8540, price2-325, price 3-27800;
Cout<<"Product        "<<setw(10) <<"Price"<<end1
      <<"Hard disk     "<<set2(10) <<price1<<end1
      <<"Mouse         "<<setw(10) <<price2<<end1
<<"Computer           "<<setw(10) <<price3<<end1;
Getch();
}
```

Output of the program

If setw manipulator is not used, the output will be

```
Product     price
Hard disk   8540
Mouse       325
Computer    27800
```

Q27. Describe the commonly used header files

Answer

The commonly used header files are listed in the following table with their purpose

Header file	Purpose
Iostream.h	Provides basic input/output operations such as cin and cout
Conio.h	Stands for console input/output. It manages input/output on console based applications. Console applications take input from keyboard and display output on monitor
Math.h	Provides basic mathematical operations such as sqrt(), pow() etc
Iomanip.h	Provides manipulator functions such as setw(), setprecision() etc
Time.h	Provides date and time operations

3.4 OPERATIONS IN C++

Q28. What is meant by operator? Give me example

Answer

Operator

An operator is a symbol that tells the computer to perform a specific computing task

Example

The + operator is used to add two numbers

3.4.1 TYPES OF OPERATORS

Q29. Explain which kinds of operators are used in C++?

Answer

The following types of operators are commonly used in C++

i. Assignment operators
ii. Arithmetic operators
iii. Arithmetic assignment operators
iv. Increment/decrement operators
v. Relational operators
vi. Logical operators
vii. Ternary operators

1. **Assignment Operator**

   Assignment operator is equal sign (=). It is used to assign value of an expression to a variable

   General form

   It has the general form

   Variable= expression:

Where expression may be a constant another variable to which a value has previously been assigned or a formula to be evaluated.

Example

$$Z = x + y;$$

When this statement is executed, the values stored in variables x and y will be added and the resulting answer will be stored in variable z

2. **Arithmetic Operators**

Arithmetic operators are used to perform arithmetic operations that include addition, subtraction, multiplication, division and to find the remainder of integer division

Types of Arithmetic Operators

The types of arithmetic operators used in C++ programming are described with their operations in the given table

Operators	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo operator

The modulo operator (%) gives the remainder after division of one number by another

Example

A = 20 % 6

Will give the result 2 which will be assigned to the variable a because 6 will divide 20 by 3 with a remainder of 2

Examples of Modulo Operator

Some more examples of modulo operator are given below

13 % 4	Will give the result 1
15 % 4	Will give the result 3
10 % 5	Will give the result 0
4 % 5	Will give the result 4

3. **Arithmetic Assignment Operators**

   In addition to equal (=) assignment operator, there are a number of assignment operators unique to C++ which are known as arithmetic assignment operators. These include +=, -=, /= and %=

Explanation

Suppose op represents an arithmetic operator. Then, the arithmetic assignment operator has the following general form to assign value of an expression to a variable

Variable op = expression;

Example

a+ = b;

It is equivalent to

a = a + b;

The effect is exactly the same but the expression is more compact

Arithmetic Assignment Operators with their Operations

The arithmetic assignment operators are described in the following table

Operator	Operation	Example
+=	Operand and assigns the result to the left side	A+=b

		It is equivalent to a=a+b operand
-=	Subtracts the right side operand from the left side operand and assigns the result to the left	K-=n It is equivalent to k=k-n side operand
*=	Multiplies the right side operand with the left side operand and assigns the result to the left	Prod*=n It is equivalent to prod=prod*n side operand
/=	Divides the left side operand by the right side operand and assigns the result to the left side	n/=m it is equivalent to n=n/m operand
%=	Takes modules and assigns the result to the left side operand	X%=y It is equivalent to x=x%y

Relational Operators

Relational operators are used to compare two values of the same type. These operators are very helpful in computer programming when a flow of program is based on a condition. After evaluation of a relational expression, the result produced in True or False

Types of Relational Operators

Six types of relational operators are available in C++ language. These are described in the following table

Operator	Operations	Example

==	Equal to	A==b
!=	Not equal to	N!=m
<	Less than	A<b+c
>	Greater than	Z<y
<=	Less than or equal to	Z<=(x+y)/2
<=	Greater than or equal to	F>=a+12

Examples of Relational Operators

The following are some examples of relational operators

x>y

x = = y

z > x+ y

z<= x + y

x! = 10

i. If x has the value 12 and y has the value 7, then the condition in first line will become true since 1 Is greater than 7
ii. The condition of second expression will become false since 12 is not equal to 7
iii. In the third expression, if z has the value 15, then the condition will become false since 15 is not greater than the sum of x and y which is 19
iv. In the fourth expression 15 Is less than 19. Therefore, the condition will become true
v. In the last expression, if x is any number other than 10 then the expression will be true it will only be false when x is equal to 10

NOTE: In C++ language, true is represented by the integer 1 & false is represented by the integer 0.

4. **Logical Operators**

   Logical operators are used in programming when it required to take some action based on more than one condition

Compound Condition

When two or more conditions are combined, it is called compound condition.

Q30. Explain the types of logical operators

Answer

Types of logical Operators

There are three types of logical operators. These are described below

Operator	Operation
&&	AND
!!	OR
!	NOT

i. **Logical AND (&&) Operator**

It is used to form compound condition in which two relational expressions are evaluated. One relational expression is to the left and the other to the right of the operator. If both of the relational expressions (conditions) are true then the compound condition is considerable true otherwise it is false

Explanation with examples

1. The following is a compound condition that uses the && operator

   (x>=1) && (x<=10)

   When the compound condition is evaluated, it will produce the result true if x is greater than or equal to 1 and less than or equal to 10. In other words, the result

will be true if both conditions are true that is x is in between 1 and 10 otherwise it will be false

2. The following compound condition will check whether the character stored in character variable ch is a lowercase letter or not

(ch<='a') && (ch<='z')

Truth table of AND logical Operator

The following truth table shows all the possible results of AND (&&) logical operator for two expressions

Expression 1	Expression 2	Expression 1 && Expression 2
False	False	False
False	True	False
True	False	False
True	True	True

ii. Logical OR (I I) Operator

Logical OR operator is also used to form a compound condition. Just like the logical AND operator one relational expression is to the left and the other to the right of the OR operator. The compound condition is true if either of the conditions is true or both are true. It is considered false only if both of the conditions are false.

Explanation

The following is an example of compound condition that uses II operator

(x>y) I I (z==8)

This compound condition will produce the result true. If one of the conditions is true, that is, if x is greater than y or z Is equal to 8. The result will only be false when both of the conditions are false, that is, x is not greater than y and z is not equal to 8

Truth table of OR logical Operator

The following truth table shows all the possible results of OR (II) logical operator for two expressions

Expression 1	Expression-2	Expression-1 I I Expression-2
False	False	False
False	True	False
True	False	False
True	True	true

iii. **Logical NOT (!) Operator**

The logical NOT operator is used with a single expression (condition) and evaluated to true if the expression is false and vice versa. In other words, it reverses the result of a single expression.

Example

The expression

!(z>=10)

Will be true if z is not less than 10. In other words, the condition will be true if z is greater than or equal to 10. Some programmers may prefer to write the above expression as given below which is easy to understand and has the same effect.

Truth table of NOT logical Operator

The following truth table shows all the possible results of NOT (!) logical operator for one expression

Expression	!Expression
False	True
True	False

Q31. Explain ternary operator OR conditional operator with examples

Answer

Ternary Operator (? :) / Conditional Operator

The ?: operator is known as ternary or conditional operator. It returns one of two values depending on the result of a condition. Therefore, it is also known as conditional operator. It is very useful in situation where the programmer needs to choose one of two options depending on a single condition

General Form

The general form of ternary operator is

Condition Expression 1 Expression2:

The condition is evaluated if it is true then Expression1 is evaluated otherwise Expression2 is evaluated

Example

The following is an example of using ternary operator

(x>y)? x+y: x-y

Explanation

When this code is executed, the condition x>y is evaluated. If the result is true then the value x+y is evaluated otherwise the value x-y is evaluated. It allows to execute different code based on the result of condition x>y

The result of ternary operator can be assigned to a variable as shown below

K= (x>y)? x+y: x-y:

Program 1

The ternary operator is demonstrated in the following program

```
#include<iostream.h>
```

```
#include<conio.h>
Void main()
{
Int x,y,k;
X=15;
Y=10;
K=<x>y)? x + y: x - y;
Cout<<"the value of k Is "<<L<<end1;
Getch();
}
```

Explanation

The value of x is greater than y. therefore, the condition (x>y) is true and k will be assigned the sum of x and y

Output of the Program

The output of the program will be

The value of k is 25

The ternary operator also allows to output text as shown below

```
(x>y)?  cout<<"x is greater than y": cout<<"x is smaller than y";
```

Program 2

This is demonstrated in the following program

```
#include<iostream.h>
#include<conio.h>
Void main()
```

```
{
Int x,y,k;
X=3;
Y=10;
(x>y) cout<"x is greater than y"<<End1;
Cout<<"x is smaller than y"<<end1;
Getch();
}
```

Explanation

In this program the value of x is smaller than y. therefore, the condition (x>y) is false and the second output statement will be executed.

Output of the Program

The output of the program will be

X is smaller than y

Q32. Explain increment and decrement operators with examples

Answer

Increment and Decrement operators

The increment operator is ++ and it is used to add one to the value stored in a variable.

The decrement operator is – and it subtracts one from the value stored in a variable.

The purpose of using these operators is simply to shorten the expression.

Examples

++x and x++	Are both equivalent to	X = x + 1

- And x --	Are both equivalent to	X = x - 1

Prefix & Postfix

When increment or decrement operator is written before the variable, it is known as prefix and when it is written after the variable, it is known as postfix

In certain situations ++x and x++ have different effect. The is because ++x increments x before using it value whereas x++ increments x after its value is used

Example

Suppose x has the value 3. The statement

Y = ++x;

Will first increment x and then assigns the value 4 to y, but the statement

Y= x++;

Will first assign the value 3 to y and then increments x to 4. In both cases, x is assigned the value 4 but y is assigned different value. The same rules applies to –x and x—as well

Example of Prefix Increment

```
#include<iostream.h>

#include<conio.h>

Void main()

{

Int x,y;

Y=10;

X=++y;

Coul<<"x: "<<x;

Coul<<"y: "<<y;
```

```
Getch():
}
```

Output of the Program

```
x: 11
y: 11
```

example of Postfix Increment

```
#include <iostream.h>
#include <conio.h>
Void main
()
{
Int x,y;
Y=10;
X=y++;
Cout<<"x:  "<<x;
Cout<<"y:  "<<y;
Getch();
}
```

Output of the Program

```
X:  10
Y:  11
```

3.4.2 UNARY, BINARY AND TERNARY OPERATORS

Q33. Explain the unary, binary and ternary operators

Answer

There are three types of operators in C++ which are unary, binary and ternary operators

Unary Operator

The operator that works on a single operand is known as unary operators. Unary operators are (-, ++, -) and the logical operator (!)

Examples of Unary Operator

Some examples of unary operator are

```
A=-b;
K++;
--x;
```

Binary Operators

The operators that work on two operands are known as binary operators. Binary operators are -, +, *, /, % and logical operators && and ||

Examples of Binary Operators

Some examples of binary operators are

```
A=b+c;
Z=x*Y;
K=d%e;
```

Ternary Operator

The conditional operator (?) is known as ternary operator since it has three parts. The three parts are a condition and two expressions. The conditions is evaluated and based on its result one of the two expressions is executed.

3.4.3 ORDER OF PRECENEDENCE OF OPERATORS

Q34. Describe the order of precedence of operator

Answer

Order of Precedence of Operators

Order of Precedence of operators describes the rules according to which operations are to be performed in an expression

In the following table, the operator that has the highest precede is written at the top and the one with the lowest precedence is written at the bottom

Precedence	Operator	Description
1.	*, /, %	Multiplication, division and remainder
2.	+, -	Addition and subtraction
3.	<, <=, >, >=	Relational operators
4.	= =, !=	Equal to and not equal to
5.	!	Logical not
6.	&&	Logical AND
7.	!!	Logical OR
8.	=, *=, /=, %=, +=, -=	Assignment Operators

Example

In the expression

7 + 3 * 2

Multiplication operator has higher precedence than the addition operator and thus will be evaluated first

Parenthesis are used in expression to change the order of evaluation of operators specified by operator precedence

Example

In the expression 7+3*2 if it is required to perform addition before multiplication then parenthesis can be used as shown below

(7 + 3) * 2

When two operations of same precedence occur in an expression then they are evaluated from left to right as they occur

When an expression contains arithmetic, relational and logical operators, the arithmetic operators are evaluated first, relational operators next and logical are evaluated last. Assignment operators are always applied at the end

3.4.4 EXPRESSIONS IN C++

Q35. Explain the expressions in C++.

Answer

Expression

An expression is a combination of constants, variables and operators. Constants and variables are operands and operators tell the computer what types of action to perform on the operands.

Types of Expressions

There are three types of expressions in C++. These are as follows

i. Arithmetic expression

ii. Relational expression

iii. Compound or logical expression

1. **Arithmetic Expression**

   An expression that contains constants, variables and arithmetic operators is called arithmetic expression

   Example

   In the expression

   A + 2 * b

   a, b and 2 are operands and + and * are arithmetic operators. When this expression is evaluated, the values stored in variables a and b are substituted and the result produces another value

   Examples

   The following are some examples of arithmetic expression

   5 * (a b)

   (a + b)/(a b+ a * b

   3 * a + 8 - b + (b + c)/2

2. **Relational Expressions**

   An expression that contains a relational operator to compare values of same type is called relational expression. Relational expressions are used in programing to create condition based on which computer takes different path during program execution

3. **Compound Expression**

   An expression that combines two or more conditions using logical operators && or II is called compound expression

   Example

   The following is an example of compound expression

   ((ch>='a') && (ch<='z') II ((ch>='A') && (ch<='Z'))

Explanation

The first compound condition ((ch>='a') && (ch<='z') || ((ch>='A') && (ch<='Z')) checks whether the character stored in variable ch is a lower case letter or not and the second compound condition ((ch>='A')&&((ch<='Z') checks whether it is an upper case letter or not if one of the two compound conditions is true, the compound expression will be true because the two compound conditions are combined with an || (OR) operator.

KEY POINTS

- A computer program is a set of instructions to perform a specific task
- Header files contain information that is required by the program in which they are used. It has .h extension
- Reserved words are special words which are reserved by a programming language for specific purpose In program
- In C++ semicolon is a statement terminator. It marks the end of a statement. All the C++ statements must end with a semi colon
- Comments are explanatory statements that help the reader in understanding source code
- A constant is a value that does not change during execution of program
- A variable is a name of memory location where data is stored. Variables are used in computer programs to store values of different data types
- Escape sequences are special characters used to control printing on the output device. These characters are not printed. These are used inside the output statement
- Order of precedence of operators describes the rules according to which operations are to be performed in an expression
- An operator is a symbol that tells the computer to perform a specific computing task

- Assignment operator is equal sign (=). It is used to assign value of an expression to a variable
- Relational operators are used to compare two values of the same type
- Logical operators are used in programming when it is required to take some action based on more than one condition
- An expression is a combination of constants, variables and operators. Constants and variables are operands and operators tell the computer what types of action to perform on the operands
- An expression that combines two or more conditions using logical operators && or II is called compound expression.