CHAPTER-6

GETTING STARTED WITH C++

TYPE A : VERY SHORT ANSWER QUESTIONS

1.	**Who was developer of C++?**
Ans.	The C++ programming language was developed at AT&T Bell Laboratories in the early 1980s by Bjarne Stroustrup.
2.	**What was the original name give to C++? Who gave the name "C++"?**
Ans.	The original name give to C++ was "C with classes". The Rick Mascitti gave the name "C++".
3.	**What is meant by token? Name the tokens available in C++.**
Ans.	The smallest individual unit in a program is known as a Token. C++ has the following tokens: 1. Keywords 2. Identifiers 3. Literals 4. Punctuators 5. Operators
4.	**What are keywords? Can keywords be used as identifiers?**
Ans.	Keywords are the reserved words having special meaning and purpose. No, keywords cannot be used as identifiers
5.	**What is an identifier? What is the identifier forming rule of C++?**
Ans.	Identifier is a name give by user for a part of the program. The first character must be a letter; the underscore (_) counts as a letter. Upper and lower-case letters are different. All characters are significant.
6.	**Is C++ case sensitive? What is meant by the term 'case sensitive'?**
Ans.	Yes, C++ is case sensitive. The term 'case sensitive' means upper and lower-case characters are treated differently.
7.	**Which of the following are valid identifiers and why/why not:** **Data_rec, _data, 1data, my.file, asm, switch, goto, break.**
Ans.	**Following are valid identifiers:** Data_rec _data, 1data **Following are invalid identifiers:** my.file contains special character asm reserved keyword switch reserved keyword goto reserved keyword break reserved keyword
8.	**What are literals? How many types of integer literals are available in C++?**
Ans.	Literals are data items that never change their value during a program run. Following three types of integer literals are available in C++: 1. Decimal Integer Constants 2. Octal Integer Constants 3. Hexadecimal Integer Constants
9.	**What is an integer constant? Write integer forming rule of C++.**
Ans.	Integer constants are whole numbers without any fractional part. Integer forming rule of C++ is as following: An integer constant must have at least one digit and must not contain any decimal point. It may contain either + or – sign. A number with no sign is assumed to be positive. Commas cannot appear in an integer constant.
10.	**How many types of integer constants are allowed in C++? How are they written?**
Ans.	In C++, an integer constant can be represented in 3 ways: 1. Decimal integer constant: In which the integer constant consist of a sequence of digit but it does not begin with digit 0. For example, 1234. 2. Octal integer constant: In which the integer constant begins with digit 0. For example, 9 can be written as 011. 3. Hexadecimal integer constant: In which the integer begins with ox or OX. For example, 11 can be written as OXB.
11.	**What kind of program elements are the following?** **13, 'a', 4.38925, "a", main()**
Ans.	13 = Decimal integer constant 'a' = Character constant 4.38925 = Floating constant "a" = String literals main() = Function
12.	**What kind of program elements are the following?**

	14, 011, 0X2A, 17, 014, 0XBC1
Ans.	14 = Decimal integer constant 011 = Octal integer constant 0X2A = Hexadecimal integer constant 17 = Decimal integer constant 014 = Octal integer constant 0XBC1 = Hexadecimal integer constant
13.	**What is a character constant in C++?**
Ans.	A character constant is one character enclosed in single quotes, as in ‘a’.
14.	**How are nongraphic characters represented in C++?**
Ans.	The nongraphic characters are represented in C++ by using escape sequences. An escape sequence is represented by a backslash (\) followed by one or more characters. For example, \n is used for newline or linefeed.
15.	**Why are characters \, ‘, “ and ? typed using escape sequences?**
Ans.	The characters \, ‘, “ and ? carry a special meaning and have a special purpose, so, if these are to be typed as it is, then escape sequences should be used.
16.	**Which escape sequences represent the newline character & null character? An escape sequence represents how many characters?**
Ans.	The ‘\n’ represents newline character and ‘\0’ represents null character. An escape sequence represents a single character.
17.	**Which of the following are valid character constants in C++?** **‘c’, ‘abc’, “abc”, my, ‘main’s’, ‘ “ ‘, ‘char, ‘\\’**
Ans.	Following are the valid character constants in C++: ‘c’, ‘\\’, ‘ ” ’
18.	**What is meant by a floating point constant in C++? How many ways can a floating constant be represented into?**
Ans.	Floating constants are numbers having fractional parts. These may be written in one of the two forms called fractional form or the exponent form.
19.	**Write the following real constants into exponent form: 23.197, 7.214, 0.00005, 0.319**
Ans.	(i) 23.197 = 0.23197 x 10^2 = 0. 23197E02 (ii) 7.214 = 0.7214 x 10^1 = 0.7214E01 (iii) 0.00005 = 0.5 x 10^{-4} = 0.5E - 04 (iv) 0.319 = 0.0319 x 10^1 = 0.0319E01
20.	**Write the following real constants into fractional form: 0.13E04, 0.417E-04, 0.4E-05, 0.123E02**
Ans.	(i) 0.13E04 = (ii) 0.417E-04 = (iii) 0.4E-05 = 0.000004 (iv) 0.123E02 = 12.3
21.	**What are string-literals in C++? What is the difference between character constants and string literals in terms of size?**
Ans.	A string literal is a sequence of characters surrounded by double quotes. The size of character constant is must be 1 character where as the size of a string literal is the number of characters plus one for terminator ‘\0’.
22.	**Which character is automatically added to strings in C++?**
Ans.	The special character ‘\0’ is automatically added to strings in C++.
23.	**What is the significance of null (\0) character in a string?**
Ans.	Each string is by default added with a special character ‘\0’ which makes the end of a string. ‘\0’ is a single character. Thus the size of a string is the number of characters plus one for this terminator.
24.	**What header file must you include with your source file to use cout and cin?**
Ans.	The header file iostream.h we must include with our source file to use cout and cin.
25.	**Name the I/O library of C++.**
Ans.	The I/O library of C++ is iostream.h
26.	**Name the standard input, output and error devices.**
Ans.	The standard input device is keyboard, the standard output and error device is screen/monitor.

27.	**How are files actually treated in C++?**
Ans.	The files are actually treated as streams of bytes in C++.
28.	**How is data at user level different from that of data at implementation level?**
Ans.	At implementation level the notion of data type is missing whereas at user level the notion of data type is present.
29.	**Who is responsible for interface between user level and lowest implementation level?**
Ans.	I/O library is responsible for interface between user level and lowest implementation level
30.	**Which component of C++ standard library predefines a set of operations for handling built-in data types?**
Ans.	The I/O library predefines a set of operations for handling built-in data types.
31.	**What are predefined stream objects in C++?**
Ans.	Following are predefined stream objects in C++: 1. cin – for standard input. 2. cout - for standard output. 3. cerr - for standard error.
32.	**If the file iostream.h is not included in a program, what happens?**
Ans.	If the file iostream.h is not included in a program, each reference to cin, cout or cerr will be flagged as a type error by the compiler.
33.	**Why is function main() special? What would happen if main() is not present in the program?**
Ans.	The main() function is the point by where all C++ programs begin their execution and continues by sequentially executing the statements within main(). A program terminates normally following execution of the last statement of main(). If main() is not present in the program, the program does not get execute.
34.	**What kinds of program errors can you encounter during programming? Why do these occur?**
Ans.	Three kinds of program errors can you encounter during programming as given below: 1. Syntax Errors: Occurs when rules of a programming language are violated. 2. Semantics Errors: Occurs when program statements are not meaningful. 3. Type Errors: Occurs when data/value of unexpected type is passed or input.
35.	**Apart from pointing out errors, what is the other role of a compiler?**
Ans.	Apart from pointing out errors, a compiler translates the corrected program text into object or assembly instruction text understood by the computer.
36.	**What do you understand by 'code generation'? Can a program be executed before it?**
Ans.	After removal of all errors in a program, compiler compiles the program to translate the program text in object or assembly instruction set which is understood by the computer. This process of translation is called code generation.
37.	**What is the input operator ">>" and output operator "<<" called?**
Ans.	The input operator ">>" is called stream extraction operator and output operator "<<" is called stream insertion operator.
38.	**What is the difference between run-time error and syntax error? Give one example of each.**
Ans.	A run-time error is occurs during the execution of a program. For example, if a program is trying to open a file which does not exist, it results into an execution error. Whereas a syntax error occur when rules of a programming language are misused. For example, if we write following statement: int a, b: then the compiler produces a syntax error as the statement is terminated by : rather than ;.

TYPE B : SHORT ANSWER QUESTIONS

1.	**What are tokens in C++? How many types of tokens are allowed in C++? Explain your answer.**
Ans.	Token is a smallest individual unit in a program. In C++ following types of tokens are allowed: **1. Keyword:** Reserve word having special meaning and purpose. For example, break, for, int etc. **2. Identifier:** Name given by user for a part of the program. For example, MyFile, _CHK etc. **3. Literals:** Data items that never change their value during a program run. For example, 1234, 'A', 2.0 etc. **4. Punctuators:** The characters which are used as punctuators in C++: [ ] ( ) { } , ; : * … = # **5. Operators:** Operators are tokens that trigger some computation or action when applied to variables ad other objects in an expression. For example, +, -, &, * etc.

2. **How are keywords different from identifiers?**

Ans. Keyword is a special word that has a special meaning and purpose. Keywords are reserved and are a few. For example, goto, switch, else etc. are keywords.
Identifier is the user-defined name given to a part of a program viz. variable, object, function etc. identifiers are not reserved. These are defined by the user but they can have letters, digits and a symbol underscore. They must begin with either a letter or underscore. For instance, _chk, chess, trial etc. are identifiers in C++.

3. **Identify the errors in the following code segments:**

```
(i) int main() { cout<<"Enter two numbers";
    cin>>num>>auto;
    float area=Legth*breadth; }

(ii) #include<isostream.h>
     void Main()
     {     int a,b;
        cin<< >>;
      if(a>b)MAX=a
     }
```

Ans. (i) 1. The header file iostream.h has not been included.
2. The variables num, Length and breadth have not been declared before using them.
3. auto is a keyword which cannot be used as a variable name.

(ii) 1. There is a spelling mistake in declaration of header file iostream.h
2. The variables MAX has not been declared before using it.
3. The statement cin<< >> has not been written correctly.
4. The statement MAX = a has not been terminated with ;.

4. **What are literals in C++? How many types of literals are allowed in C++?**

Ans. Literal mean constant i.e., the data items that never change their value during a program run.
C++ allows four types of literals:
(i) Integer literal (ii) Character literal (iii) Floating Point literal (iv) String literal

5. **How are integer constants represented in C++? Explain with examples.**

Ans. In C++, an **integer constants can be represented in 3 ways;**
1. As a decimal integer constant in which the integer constant consisting of a sequence of digits is taken to be decimal integer constant unless it begins with 0 (digit zero). For example, 1234 is a decimal integer constant but 01234 is not a decimal integer constant.
2. As an octal integer constant in which the integer constant begins with 0. For example, decimal integer 9 will be written as 011as octal integer.
3. As a hexadecimal integer constant in which the integer constant begins with ox or OX. For example, decimal 11 will be written as 0XB as hexadecimal integer.

6. **What are character constants in C++? How are these implemented?**

Ans. A character constant is one or more characters enclosed in single quotes, as in 'z'. Single character constants e.g., 'c' or 'A' have type char which is a C++ data type for characters. The value of a single character constant is the numerical value of the character in the machine's character set. For instance, the value of 'c' will be 99 which is ASCII value of c and the value of 'A' will be 65 which ASCII value of A. multi-character constant have type int, a C++ data type for integers. The value of a multicharacter constant is implementation dependent.

7. **Can nongraphic characters be used and processed in C++? How? Give examples to support your answer.**

Ans. Yes, in C++ nongraphic characters can be used and processed. Nongraphic characters are those characters that cannot be typed directly from keyboard e.g., backspace, tabs, carriage return etc. these nongraphic characters can be represented by using escape sequences. An escape sequence is represented by a backslash (\0 followed by one or more characters. For example, '\n' is used for newline or linefeed, '\t' is used for horizontal tab.

8. **How are floating constants represented in C++? Give examples to support your answer.**

Ans. In C++, floating constants are represented in following two ways:
1. Fractional form: A floating constant in fractional for must have at least one digit before a decimal point and at

least one digit after the decimal point. It may also have either + or – sign preceding it. A real constant with o sign is assumed to be positive. For example, 2.0, -13.0, -0.00625

2. Exponent form: A floating constant in exponent form has two parts: a mantissa and an exponent. The mantissa must be either an integer or a proper real constant. The mantissa is followed by a letter E or e and the exponent. The exponent must be an integer. For example, 152E05, 0.152E08, 152E+8, -0.172E-3

9. How are string-literals represented and implemented in C++?

Ans. A string literal is a sequence of characters surrounded by double quotes. Each string is by default added with a special character '\0' which makes the end of a string. '\0' is a single character. Thus the size of a string is the number of characters plus one for this terminator. For example, "abc" size is 4.
Thus "abc" will actually be represented as "abc\0" in the memory i.e., along with the terminator character.

10. What are operators? What is their function? Give examples of some unary and binary operators?

Ans. Operators are tokens that trigger some computation or action when applied to variables ad other objects in an expression. Following are some unary and binary operators:

Unary Operators		Binary Operators	
&	Address operator	+	Addition
*	Indirection operator	-	Subtraction
++	Increment operator	%	Reminder/Modulus
--	Decrement operator	&&	logical AND

11. Why is it important to include iostream.h in C++ programs?

Ans. The header file iostream.h is included in every C++ program to implement input/output facilities. Input/output facilities are implemented through a component of C++ standard library, iostream.h which, is I/O library. Declarations and functions of cin, cout and cerr are contained within iostream.h. If the file iostream.h is not included in a program, each reference to cin, cout or cerr will be flagged as a type error by the compiler.

12. How are files and devices implemented at the lowest level?

Ans. In C++, all devices are treated as files. Thus, the standard input device, the standard output device and the standard error device are all treated as files. At its lowest level, a file is interpreted simply as a sequence, or stream of bytes. At this level, the notion of a data type is absent i.e., data is treated simply as sequence of bytes without considering its data type.

13. What is the function of I/O library in C++?

Ans. Same as Question No. 3 of Long Answer Question.

14. What are streams in C++? What are predefined streams in C++? What are their functions?

Ans. A stream is simply a sequence of bytes. The predefine stream objects for input, output and error are as follows:
1. cin – as istream class object tied to standard input. cin stands for console input.
2. cout – an ostream class object tied to standard output. cout stands for console output.
3. cerr - an ostream class object tied to standard error. cerr stands for console error.

15. Why is main() function so important?

Ans. Every C++ program must have a function named main(). Program execution begins at main() and continues by sequentially executing the statements within main(). A program terminates normally following execution of the last statement of main(). And for the same reason, it is essential that all C++ programs have a main() sunction.

16. Find out the errors, if any, in the following C++ statements:

```
(i) cout<<"a=" a;     (ii) m=5, n=12; 0=15  (ii) cout<<"x";<<x;
(iv) cin>>y;>>j       (v) cin>>"\n">>y;     (vi) cout>>\n"abc";
(vii) a=b+c           (viii) break=x*y;
```

Ans. (i) The '<<' operator is missing.
(ii) The data type is missing, illegal use of ';' and not terminated by ;.
(iii) Illegal use of semicolon (;).
(iv) Illegal use of ';' and not terminated by ;.
(v) Illegal use of "\n" escape sequence.
(vi) The '\n' should be in double quote.
(vii) Semicolon missing.
(viii) The reserved word 'break' should not be used as variable name.

17. Write a program to display the following output using a single cout statement.

Program=20
Documentation=23
Logic=21
Flow chart=18

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      cout<<"Program = 20"<<"\nDocumentation = 23"<<"\nLogic = 21"<<"\nFlow
chart = 18";
      getch();
}
```

18. **Write a program to read values of w, x, y and z and display the values of P, where**

P = $\frac{w+x}{y-z}$

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      int w,x,y,z,P;
      cout<<"Enter w : ";
      cin>>w;
      cout<<"Enter x : ";
      cin>>x;
      cout<<"Enter y : ";
      cin>>y;
      cout<<"Enter z : ";
      cin>>z;
      P=((w+x)/(y-z));
      cout<<"P="<<P;
      getch();
}
```

19. **Write a C++ program that reads temperature in Celsius and display it in Fahrenheit.**

Ans.

```
#include<conio.h>
#include<iostream.h>
void main()
{
      float f,c;
      clrscr();
      cout<<"Enter Fahrenheit degree to find temperature in celsius: ";
      cin>>f;
      c = (f-32)/1.8;
      cout<<"\n\n\tCELSIUS DEGREE = "<<c;
      getch();
}
```

20. **Write a program that displays on screen**

a. the character 'z' **b. the name 'Mohan'** **c. the number 1977**

Using (i) single cout statement (ii) multiple cout statements.

Ans. **(i) single cout statement:**

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      cout<<"a.the character 'z'"<<"\tb.the name 'Mohan'"<<"\tc.the number
1977";
      getch();
}
```

(ii) multiple cout statements:

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      cout<<"a.the character 'z'";
      cout<<"\tb.the name 'Mohan'";
      cout<<"\tc.the number 1977";
      getch();
}
```

21. **Assuming that there are 7.481 gallons in a cubic foot, write a program that asks the user to enter the number of gallons, and then display the equivalent in cubic feet.**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      float gallon,cubic;
      cout<<"Enter the number of gallons: ";
      cin>>gallon;
      cubic=gallon/7.481;
      cout<<"\n Cubic feet= "<<cubic;
      getch();
}
```

22. **Write a program to generate the following table:**

1992	**17421**
1993	**29210**
1994	**100523**
1995	**106802**
1996	**127000**

Use a single cout statement for output. (Hint: Make use of \n and \t.)

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      cout<<"\t1992\t\t17421\n\t1993\t\t29210\n\t1994\t\t100523\n\t1995\t\t1068
02\n\t1996\t\t127000\n ";
      getch();
}
```

23. **Write a program that generate the following output:**

```
5
10
9
```

Assign value 5 to a variable using =, Multiply it with 2 to generate 10 and subtract 1 to generate 9.

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      int a=5;
      cout<<a;
      a=a*2;
      cout<<"\n"<<a;
      cout<<"\n"<<a-1;
      getch();
}
```

24. **Write a C++ program that accepts radius of a circle and prints its area.**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      float r,area;
      cout<<"Enter radius: ";
      cin>>r;
      area=2*3.14*r;
      cout<<"\narea= "<<area;
      getch();
}
```

25. **Write a C++ program to print the output mentioned in question 23 in different lines i.e., all values must appear in the different lines.**

Ans. **Code will be same as answer of question 23**

26. **Write a C++ program that accept marks in 5 subjects and output average marks.**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      float m1,m2,m3,m4,m5,avg;
      cout<<"Enter marks obtained in 5 subjects : ";
      cin>>m1>>m2>>m3>>m4>>m5;
      avg=(m1+m2+m3+m4+m5)/5;
      cout<<"\nAverage= "<<avg;
      getch();
}
```

27. **Write a C++ program to accept two integers and print their sum.**

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
      clrscr();
      float a,b,sum;
      cout<<"Enter a and b ";
      cin>>a>>b;
      sum=a+b;
      cout<<"\nSum= "<<sum;
      getch();
}
```

TYPE C : LONG ANSWER QUESTIONS

1. **Explain tokens in C++. Also discuss their role and importance.**

Ans. **Token:** The smallest individual unit in a program is known as a Token. C++ has the following tokens:

Keywords, Identifiers, Literals, Punctuators and Operators.

	1. Keywords: The keywords are the words that convey a special meaning to the language compiler. **2. Identifier:** An identifier is the name given by user for a unit of the program. **3. Literals:** Literals are data items that never change their value during a program run. C++ allows several kinds of literals as following: (i) integer-constant (ii) character-constant (iii) floating-constant (iv) string-literal **4. Punctuators:** The characters which are used as punctuators in C++: [ ] ( ) { } , ; : * … = # that enhance a program's readability ad give proper meaning to statements, expressions etc. as per syntax. **5. Operators:** Operators are tokens that trigger some computation or action when applied to variables ad other objects in an expression.
2.	**Explain different types of literals in C++ by giving appropriate examples.**
Ans.	**Literals:** Literals are data items that never change their value during a program run. C++ allows several kinds of literals as following: (i) integer-constant (ii) character-constant (iii) floating-constant (iv) string-literal **<u>Integer Constant:</u>** Integer constant are whole numbers without any fractional part. C++ allows three types of integer constant: (a) Decimal Integer constants: For example, 1234, +97. (b) Octal Integer constants: For example, decimal integer 8 will be written as 010 as octal integers. (c) Hexadecimal Integer constants: For example, decimal 12 will be written as 0XC as hexadecimal integer. **<u>Character Constants:</u>** A character constant is single character enclosed in single quotes. For example, 'z'. **<u>Floating Constants:</u>** Floating constants are also called real constants. Real constants are numbers having fractional parts. These may be written in one of the two forms called fractional form or the exponent form. Example of valid real constants in fractional form: 2.0, 17.5, -13.0, -0.00625 Example of valid real constants in exponent form: 152E05, 0.1523E08, 152E+ 8, 0.172E-3 **<u>String Literals:</u>** Multiple Character constants, if enclosed in double quotes; these are treated as string-literals. Each string literal is automatically added with a terminating character '\0\. For example, "abc".
3.	**Explain the role and importance of file iostream.h in C++.**
Ans.	The header file iostream.h is included in every C++ program to implement input/output facilities. Input/Output facilities are not defined within C++ language, but rather are implemented through a component of C++ standard library, iostream.h which is I/O library. The functions of iostream.h: 1. At the lowest implementation level, where the notion of data type is missing and files are treated as streams of bytes, the I/O library manages the transfer of these bytes. 2. at the user level, where the notion of data type is present, I/O library manages the interface between these two levels i.e., between user level and the lowest implementation level. 3. the I/O library predefines a set of operations for handling reading and writing of built-in data types.
4.	**Explain the role of a compiler in a programming language. What type of errors it generally reports?**
Ans.	A part of the compiler's job is to analyze the program code for "correctness". If the meaning of a program is correct, then a compiler cannot detect errors, but a compiler can certainly detect errors in the form of a program. A compiler generally reports following types of errors: 1. Syntax Errors: Occurs when rules of a programming language are violated. 2. Semantics Errors: Occurs when program statements are not meaningful. 3. Type Errors: Occurs when data/value of unexpected type is passed or input. 4. Run-time Errors: Occurs during the execution of a program. 5. Logical Errors: Causes a program to produce incorrect output. A compiler reports an error by flashing an error message. An error message contains a line number and a brief description of the error. The second thing, a compiler does is, it translates the corrected program text into object or assembly instruction text understood by the computer. A compiler can report only the syntax errors, semantics errors, and type errors. Run-time errors become know at run-time and logical errors can be detected by tracing the program line by line and checking its output.