Unit - III CHAPTER - 9 INTRODUCTION TO C++ PART – I Choose the correct answer. 1. Who developed C++? (a) Charles Babbage (b) **Bjarne Stroustrup** (c) Bill Gates (d) Sundar Pichai 2. What was the original name given to C++?(a) CPP (b) Advanced C (c) C with Classes (d) Class with C 3. Who coined C++? (a) Rick Mascitti (b) Rick Bjarne (c) Bill Gates (d) Dennis Ritchie 4. The smallest individual unit in a program is: (a) Program (b) Algorithm (c) Flowchart (d) Tokens 5. Which of the following operator is extraction operator of C++?(d) ^^ (b) << (a) >> $(c) \Leftrightarrow$ 6. Which of the following statements is not true? (a) Keywords are the reserved words convey specific meaning to the C++ compiler. (b) Reserved words or keywords can be used as an identifier name. (c) An integer constant must have at least one digit without a decimal point. (d) Exponent form of real constants consists of two parts 7. Which of the following is a valid string literal? (b) 'Welcome' (a) 'A' (c) 1232 (d) "1232" 8. A program written in high level language is called as (a) Object code (b) Source code (c) Executable code (d) All the above 9. Assume a= 5, b=6; what will be result of a&b? (a) 4 (b) 5 (c) 1 (d) 010. Which of the following is called as compile time operators? (a) sizeof (b) pointer (c) virtual (d) this

Part – II

Answer to all the questions

1. What is meant by a token? Name the token available in C++.

The smallest individual unit in a program is known as a "Token" or "Lexical unit."

C++ has the following tokens:

Keywords, Identifiers, Constants, Operators, Punctuators.

- 2. What are keywords? Can keywords be used as identifiers?
- \blacktriangleright Keywords are the reserved words which convey specific meaning to the C++ compiler.
- ➢ Keywords are Essential elements of C++ programs.
- \succ C++ is a case sensitive programming language so, all the keywords must be in lowercase.

 $\checkmark\,$ Reserved words or keywords cannot be used as an identifier name.

3. The following constants are of which type?

Decimal (Integer)
Octal (Integer)
Hexadecimal (Integer)
Real constant (Floating Point)

4. Write the following real constants into the exponent form:

(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

5. Assume n=10; what will be result of n>>2;?

n = 10 (00001010)₂

А	В
(a) Modulus	(1) Tokens
(b) Separators	(2) Remainder of a division
(c) Stream extraction	(3) Punctuators
(d) Lexical Units	(4) get from

Part – III

Answer to all the questions

1. Describe the differences between keywords and identifiers?

KEYWORDS	IDENTIFIERS
Keywords are the reserved words which convey specific meaning to the C++ compiler.	Identifiers are the user-defined names given to different parts of the C++ program
Keywords are the essential elements to	These are the fundamental building
construct C++ programs.	blocks of a program.
EX: int, void, break, do, if etc	EX: variable_name, function_name,
	array_name, classe_name etc

- 2. Is C++ case sensitive? What is meant by the term "case sensitive"?
 - ➤ C++ is a case sensitive programming language.
 - \triangleright C++ is case sensitive as it treats upper and lower-case characters differently.
 - ➤ Capital Letters ≠ Small Letters
- 3. Differentiate "=" and "==".
 - = **Equal (Assignment Operator)** Assign a value of an Variable.
 - = = Equal to (Equality Operator) To Indicate Two Operands are Equal.
- 4. Assume a=10, b=15; What will be the value of a^b?

 $a=10 = (00001010)_2$ $b=15 = (00001111)_2$ $a^b = (00000101)_2 = (5)_{10}$

5. What is the difference between "Run time error" and "Syntax error"?

Run time error	Syntax error
A run time error occurs during the execution	Syntax is a set of grammatical rules to
of a program.	construct a program.
It is occurs because of some illegal operation	Syntax errors occur when grammatical rules
that takes place.	of C++ are violated.
EX: if a program tries to open a file which does not exist, it results in a run-time error .	EX: cout << "Welcome to Programming in C++" Error: End of the above statement ; is missing.

6. What are the differences between "Logical error" and "Syntax error"?

Logical error	Syntax error
A Program has not produced expected result even though the program is grammatically correct.	Syntax is a set of grammatical rules to construct a program.
Program is grammatically correct, but it contains some logical error.	Syntax errors occur when grammatical rules of C++ are violated.
Semantic error is also called as "Logic Error".	EX: cout << "Welcome to Programming in C++" Error: End of the above statement ; is missing.

7. What is the use of a header file?

- If you fail to include iostream in your program, an error message will occur on cin and cout; and we will not be able to get any input or send any output.
- iostream header file contains the definition of its member objects cin and cout.

8. Why is main function special?

- C++ program is a collection of functions. Every C++ program must have a main function.
- The main() function is the starting point where all C++ programs begin their execution.
- Therefore, the executable statements should be inside the main() function.
 i.e The main function always special.

9. Write two advantages of using include compiler directive.

- ➤ All C++ programs begin with include statements starting with a #.
- > The symbol # is a directive for the preprocessor.
- > Statements are processed before the compilation process begins.

10. Write the following in real constants.

(i)	15.223		$1.5223 \text{ X}10^{1}$	→ 1.5223E-1
(ii)	211.05		2.1105×10^2	→ 2.1105E-2
(iii)	0.00025	>	25 X 10 ⁻⁵	→ 25E-5

Part – IV

1. Write about Binary operators used in C++.

Binary Operators - Require two operands C++ Operators are classified as:

- (1) Arithmetic Operators
- (2) Relational Operators
- (3) Logical Operators
- (4) Bitwise Operators
- (5) Assignment Operators
- (6) Conditional Operator

(1) Arithmetic Operators

Arithmetic operators to perform simple arithmetic operations like addition, subtraction, multiplication, division etc.,

Operator	Operation	Example
+	Addition	10 + 5 = 15
-	Subtraction	10 - 5 = 5
*	Multiplication	10 * 5 = 50
/	Division	10 / 5 = 2 (Quotient of the division)
%	Modulus (To find the reminder of a division)	10 % 3 = 1(Remainder of the division)

(2) Relational Operators

Relational operators are used to determine the relationship between its operands. When the relational operators are applied on two operands, the result will be a Boolean value i.e **1** or **0** to represents **True** or **False** respectively.

C++ provides six relational operators. They are,

Operator	Operation	Example
>	Greater than	a > b
<	Less than	a < b
>=	Greater than or equal to	a >= b
<=	Less than or equal to	a <= b
==	Equal to	a == b
!=	Not equal	a != b

(3)Logical Operators

A logical operator is used to evaluate logical and relational expressions. The logical operators act upon the operands that are themselves called as logical expressions.

C++ provides three logical operators.

Operator	Operation	Description
&&	AND	The logical AND combines two different relational expressions in to one. It returns 1 (True), if both expression are true, otherwise it returns 0 (false).
	OR	The logical OR combines two different relational expressions in to one. It returns 1 (True), if either one of the expression is true. It returns 0 (false), if both the expressions are false.
!	NOT	NOT works on a single expression / operand. It simply negates or inverts the truth value. i.e., if an operand / expression is 1 (true) then this operator returns 0 (false) and vice versa

Example: a = 5, b = 6, c = 7;

Expression	Result
(a <b) &&="" (b<c)<="" td=""><td>1 (True)</td></b)>	1 (True)
(a>b) && (b <c)< td=""><td>0 (False)</td></c)<>	0 (False)
(a <b) (b="" ="">c)</b)>	1 (True)
!(a>b)	1 (True)

(4)Bitwise Operators

Bitwise operators work on each bit of data and perform bit-by-bit operation. In C++, there are three kinds of bitwise operators, which are:

- (i) Logical bitwise operators
- (ii) Bitwise shift operators
- (iii) One's complement operators

5

(5)Assignment Operator:

Assignment operator is used to assign a value to a variable which is on the left hand side of an assignment statement.

= (equal) is commonly used as the assignment operator in all computer programming languages.

This operator copies the value at the right side of the operator to the left side variable. It is also a binary operator.

(6) Conditional Operator:

- ➤ In C++, there is only one conditional operator is used. **?:** is a conditional Operator.
- > This is a Ternary Operator.
- \blacktriangleright This operator is used as an alternate to if ... else control statement.



2. What are the types of Errors?

Some common types of errors are given below:

Type of Error	Description
Syntax Error	 Syntax is a set of grammatical rules to construct a program. Every programming language has unique rules for constructing the sourcecode. Syntax errors occur when grammatical rules of C++ are violated. Example: if you type as follows, C++ will throw an error. cout << "Welcome to Programming in C++" As per grammatical rules of C++, every executable statement should terminate with a semicolon. But, this statement does not end with a semicolon.
Semantic Error	• A Program has not produced expected result even though the program is grammatically correct. It may be happened by wrong use of variable / operator / order of execution etc. This means, program is grammatically correct, but it contains some logical error. So, Semantic error is also called as "Logic Error".
Run-time error	 A run time error occurs during the execution of a program. It is occurs because of some illegal operation that takes place. For example, if a program tries to open a file which does not exist, it results in a run-time error

Operator	Name of Operator
+=	Addition Assignment
-=	Subtraction Assignment
*=	Multiplication Assignment
/=	Division Assignment
%=	Modulus Assignment

3. Assume a=15, b=20; What will be the result of the following operations?

	a = 15 b = 20	\rightarrow	$\begin{array}{c} (00001111)_{\ 2} \\ (00010100)_{\ 2} \end{array}$		
(a)	a&b	=	(00000100) ₂	=	(4) ₁₀
(b)	a b	=	(00011111) ₂	=	(31)10
(c)	a^b	=	(00011011) ₂	=	(27)10
(d)	a>>3	=	(0000001) ₂	=	$(1)_{10}$
(e)	(~b)	=	(00001011) ₂	=	$(11)_{10}$

Truth table for bitwise operators	(AND, OR,	XOR)
-----------------------------------	-----------	------

A	В	A & B	A B	A ^ B
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

Shift right (>>)– The value of the left operand is moved to right by the number of bits specified by the right operand.

The bitwise One's compliment operator ~(**Tilde**), inverts all the bits in a binary pattern, that is, all 1's become 0 and all 0's become 1.

7

DATA TYPES, VARIABLES AND EXPRESSIONS

Part – I

Choose the correct answer.

1. How many categ	ories of data types availa	ble in C++?			
(a) 5	(b) 4	(c) 3	(d) 2		
2. Which of the following data types is not a fundamental type?					
(a) signed	(b) int	(c) float	(d) char		
3. What will be the	result of following stater	nent?UOUL(JU A VOG		
char ch= 'B'	?; ;				
cout << (int)) ch;				
(a) B	(b) b	(c) 65	(d) 66		
4. Which of the cha	racter is used as suffix to	indicate a floating poir	nt value?		
(a) F	(b) C	(c) L	(d) D		
5. How many bytes	of memory allocates for	the following variable	declaration if you are using Dev		
C++? short	t int x;				
(a) 2	(b) 4	(c) 6	(d) 8		
6. What is the output	at of the following snippe	et?			
char $ch = A$	κ ';				
ch = ch + 1;					
(a) B	(b) A1	(c) F	(d) 1A		
7. Which of the foll	owing is not a data type	modifier?			
(a) signed	(b) int	(c) long	(d) short		
8. Which of the foll	owing operator returns th	ne size of the data type?			
(a) sizeof()	(b) int ()	(c) long ()	(d) double ()		
9. Which operator t	o be used to access refere	ence of a variable?			
(a) \$	(b) #	(c) &	(d) !		
10. This can be used as alternate to endl command:					
(a) \t	(b) \b	(c) \0	(c) \n		

Part – II

Answers to all the questions (2 Marks):

1. Write a short note const keyword with an example.

- > Literals are data items whose values do not change during the execution of a program.
- \blacktriangleright const is the keyword used to declare a constant.

Example:

int num = 100; (The value of num can be changed during the execution.)

const int num = 100; (The value can never be changed during the execution.)

2. What is the use of setw() format manipulator?

- Setw() format manipulator is used to set the width of the given field into the output.
- > setw manipulator sets the width of the field assigned for the output.
- > The field width determines the minimum number of characters to be written in output.

3. Why is char often treated as integer data type?

- > Character data type accepts and returns all valid ASCII characters.
- Character data type is often said to be an integer type, since all the characters are represented in memory by their associated ASCII Codes.
- ➤ If a variable is declared as char, C++ allows storing either a character or an integer value.

4. What is a reference variable? What is its use?

A reference provides an alias for a previously defined variable. Declaration of a reference consists of base type and an & (ampersand) symbol;
 Usage:

▶ Reference variable name is assigned the value of a previously declared variable.

5. Consider the following C++ statement. Are they equivalent?

char ch = 67; char ch = 'C';

Yes. Above two characters(ch=67, ch='C') are equal.

6. What is the difference between 56L and 56?

56 -	is a Integer Number	-	2 Bits
------	---------------------	---	--------

56L - is a Long Integer Number - 4 Bits

7. Determine which of the following are valid constant? And specify their type.

(i) 0.5	-	Valid Floating Constant	
(ii) 'Name'	-	Invalid String Constant	(Enclosed within Double quotes)
(iii) '\t'	-	Valid Horizontal tab, But	it is not valid constant
(iv) 27,822	-	Invalid Decimal Constant	(Commas is not allowed)

Decimal

number

65

66

67

68

69 70

Alphabets

Α

В

С

D

Е

8. Suppose x and y are two double type variable that you want add as integer and assign to an integer variable. Construct a C++ statement for the doing so.

Output:

Two double type variable – double x,y and one integer variable – int z Adding x and y and assign to an integer variable z(result must be integer)

Eg: x=2.5, y=1.2 then result z=3(it must be integer)

#include <iostream.h></iostream.h>
#include <conio.h></conio.h>
void main()
{
clrscr();
int z;
double x,y;
cout << "Give input: x and y:";
cin>>x>>y;
z=x+y;
$cout \ll "Result z = " \ll z;$
getch();
}

Give input: x and y: 2.5 1.2 Result z = 3

9. What will be the result of following if num=6 initially.

(a) cout << num; $\longrightarrow 6$ (b) cout << (num==5); $\longrightarrow 0$

10. Which of the following two statements are valid? Why? Also write their result. int a;

(i) a = 3,014;(ii) a=(3,014);

Above the two statements is Invalid.

Special Symbols are not allowed in the integer values (Commas, Open and Close Brackets)

Part – III

Answers to all the questions (3 Marks):

1. What are arithmetic operators in C++? Differentiate unary and binary arithmetic operators. Give example for each of them.

The symbols which are used to do some mathematical or logical operations are called as "Operators".

(i) Unary Operators	- Require only one operand	Ex:	a ++
(ii) Binary Operators	- Require two operands	Ex:	a + b
(iii) Ternary Operators	- Require three operands	Ex:	a>b ? a : b

www.Padasalai.Net

2.

Evaluate x+= x + ++x; Let x=5;

x += 5 + (++x) x += 5 + 6 x += 11 x = x + 11x = 16

3. How relational operators and logical operators related to one another?

RELATIONAL OPERATORS	LOGICAL OPERATORS
Relational operators are used to determine	Relational operators are used to determine
the relationship between its operands.	the relationship between its Expressions.
When the relational operators are applied	When the relational operators are applied
on two operands, the result will be a	on two Expressions, the result will be a
Boolean value i.e 1 or 0 to represents True	Boolean value i.e 1 or 0 to represents True
or False respectively.	or False respectively.

4. Evaluate the following C++ expressions where x, y, z are integers and m, n are floating point numbers. The value of x = 5, y = 4 and m=2.5;

(i)
$$n = x + y/x$$
;
 $n = 5 + (4/5)$
 $n = 5 + 0.8$
 $n = 5.8$
(ii) $z = m * x + y$;
 $z = (2.5 * 5) + 4$
 $z = 12.5 + 4$
 $z = 16.5$
 $z = 16 (z - is integer)$
(iii) $z = (x++) * m + x$;
 $z = ((5++) * 2.5) + 5$
 $z = (5 * 2.5) + 5$
 $z = (5 * 2.5) + 5$
 $z = 12.5 + 5$
 $z = 17.5$
 $z = 17.(z - is integer)$

10

Evaluate Yourself-1 (Book Interior Questions)

1. What is meant by literals? How many types of integer literals available in C++?

Literals are data items whose values do not change during the execution of a program. Therefore Literals are called as Constants.

Types of integer literals:

- i. Decimal
- ii. Octal
- iii. Hexadecimal

2. What kind of constants are following?

i)	26	-	Integer
ii)	015	-	Octal
iii)	0xF	-	Hexadecimal
iv)	014.9	-	Floating

3. What is character constant in C++?

A character constant is any valid single character enclosed within single quotes. A character constant in C++ must contain one character and must be enclosed within a single quote.

Example : 'A', '2', '\$'

4. How are non graphic characters represented in C++?

non-printable characters can be represented by using escape sequences. An escape sequence is represented by a backslash followed by one or two characters.

5. Write the following real constants into exponent form:

i)	32.179	-	0.32179E-2
ii)	8.124	-	0.8123E-1
iii)	0.00007	-	0.7E4

6. Write the following real constants into fractional form:

i)	0.23E4	-	0.0023
ii)	0.517E-3	-	517
iii)	0.5E-5	-	50000

7. What is the significance of null (\0) character in a string?

Sequence of characters enclosed within double quotes are called as String literals. By default, string literals are automatically added with a special character '\0' (Null) at the end

Therefore, the string "welcome" will actually be represented as "welcome0" in memory and the size of this string is not 7 but 8 characters i.e., inclusive of the last character 0.

Evaluate Yourself-2

1. What is use of operators?

The symbols which are used to do some mathematical or logical operations are called as "**Operators**".

2. What are binary operators? Give examples arithmetic binary operators.

Binary Operators - Require two operands

- 1) Arithmetic Operators
- (2) Relational Operators
- (3) Logical Operators
- (4) Bitwise Operators
- (5) Assignment Operators
- (6) Conditional Operator

Operator	Operation	Example
+	Addition	10 + 5 = 15
-	Subtraction	10 - 5 = 5
*	Multiplication	10 * 5 = 50
/	Division	10 / 5 = 2 (Quotient of the division)
%	Modulus (To find the reminder of a division)	10 % 3 = 1(Remainder of the division)

3. What does the modulus operator % do?

Modulus (%) Operator is used to find the reminder of a division.

Example: 10 % 3 = 1

4. What will be the result of 8.5 % 2?

0.5 (*Reminder is the answer*)

5. Assume that R starts with value 35. What will be the value of S from the following expression?

S=(R--)+(++R) S=(35--)+(++35) S=35+36S = 71

6. What will be the value of j = -k + 2k. if k is 20 initially ?

- j = k + 2k j = - 20 + 2*20 j = 19 + 40j = 59
- 7. What will be the value of **p** = **p** * ++j where **j** is 22 and **p** = 3 initially?
 - p = p * ++jp = 3 * ++22
 - p = 3 * 23
 - **p** = 69

8. Give that i = 8, j = 10, k = 8, What will be result of the following expressions?

8<8 0 (i) i < k= (ii) 8<10 = 1 i < j(iii) i > = k8>=8 = 1 8 = 10 =0 (iv) i = = i(v) j ! = k 10!=8 =1

9. What will be the order of evaluation for the following expressions?
(i) i + 3 >= j - 9

$$(i + 3) >= (j - 9)$$

(ii)
$$a + 10$$

$$(a+10)$$

10. Write an expression involving a logical operator to test, if marks are 75 and grade is 'A'.

If (marks > 74 && marks < 101)

Grade is 'A'

Evaluate Yourself-3

1. What do you mean by fundamental data types?

Fundamental (atomic) data types are predefined data types available with C++. There are five fundamental data types in C++: **char, int, float, double** and **void**. Actually, these are the keywords for defining the data types.

2. The data type char is used to represent characters. then why is it often termed as an integer type? Character data type accepts and returns all valid ASCII characters.

Character data type is often said to be an integer type, since all the characters are represented in memory by their associated **ASCII Codes.**

If a variable is declared as char, C++ allows storing either a character or an integer value.

3. What is the advantage of floating point numbers over integers? floating point operations takes more time to execute compared to the integer type. ie., floating point operations are slower than integer operations. This is a disadvantage of floating point operation.

4. The data type double is another floating point type. Then why is it treated as a distinct data type?

This is for double precision floating point numbers. (precision means significant numbers after decimal point). The double is also used for handling floating point numbers. But, this type occupies double the space than float type. This means, more fractions can be accommodated in double than in float type.

13

5. What is the use of void data type?

The literal meaning for void is 'empty space'. Here, in C++, the void data type specifies an empty set of values.

> It is used as a return type for functions that do not return any value.

Evaluate Yourself-4

1. What is modifiers? What is the use of modifiers?

Modifiers are used to modify the storing capacity of a fundamental data type except void type. Usually, every fundamental data type has a fixed range of values to store data items in memory.

2. What is wrong with the following C++ statement: long float x;

Data type				
float	signed fractional value			
double	signed more precision fractional value			
long double	signed more precision fractional value			

long is not a float type.

Float only allowed three types.

They are,

- 1. float
- 2. double
- 3. long double

3. What is variable? Why a variable called symbolic variable?

- Variables are user-defined names assigned to specific memory locations in which the values are stored.
- Variables are also identifiers; and hence, the rules for naming the identifiers should be followed while naming a variable. These are called as symbolic variables because these are named locations.

4. What do you mean by dynamic initialization of a variable? Give an example.

A variable can be initialized during the execution of a program. It is known as "Dynamic initialization".

For example,

int num1, num2, sum;

sum = num1 + num2;

The above two statements can be combined into a single one as follows: int sum = num1+num2;

5. What is wrong with the following statement?

const int x;

In this statement x – does not contain any value. **Const int** data type to initialize specific value during the execution of a program, it can't be changed.

Evaluate Yourself-5

1. What is meant by type conversion?

The process of converting one fundamental type into another is called as **"Type Conversion"**.

C++ provides *two types* of conversions.

- (1) Implicit type conversion
- (2) Explicit type conversion

2. How implicit conversion different from explicit conversion?

Implicit Conversion	Explicit Conversion
An Implicit type conversion is a conversion performed by the compiler automatically.	C++ allows explicit conversion of variables or expressions from one data type to another specific data type by the programmer.
It is called as "Automatic conversion".	It is called as "type casting".
Ex:	Ex:
int a=6;	float varf=78.685;
float $b=3.14$;	cout << (int) varf;
cout << a+b;	

3. What is difference between endl and \n?

Endl - Inserts a new line and flushes the buffer (Flush means – clean)

'**n**' - Inserts only a new line.

4. What is the use of references?

- > A reference provides an alias for a previously defined variable.
- > Reference variable name is assigned the value of a previously declared variable.

5. What is the use of setprecision ()?

This is used to display numbers with fractions in specific number of digits.

Syntax:

setprecision (number of digits);

Example:

float hra = 1200.123;

cout << setprecision (5) << hra;

Extra Questions (2 & 3 Marks)

1. Who developed c++ program?

C++ is one of the most popular programming language developed by Bjarne Stroustrup at AT & T Bell Lab during 1979.

2. Why C++ is called hybrid language?

C++ supports both procedural and Object Oriented Programming paradigms. Thus, C++ is called as a **hybrid language.**

3. List any two benefits of learning C++?(Any two points)

- C++ is a highly portable language and is often the language of choice for multi-device, multiplatform app development.
- C++ is an object-oriented programming language and includes classes, inheritance, polymorphism, data abstraction and encapsulation.
- ➤ C++ has a rich function library.
- ➤ C++ allows exception handling, inheritance and function overloading.
- C++ is a powerful, efficient and fast language. It finds a wide range of applications from GUI applications to 3D graphics for games to real-time mathematical simulations.

4. Write the Rules for naming an identifier?

- > The first character of an identifier must be an alphabet or an underscore (_).
- Only alphabets, digits and underscore are permitted. Other special characters are not allowed as part of an identifier.
- ➤ C++ is case sensitive as it treats upper and lower-case characters differently.
- ➢ Reserved words or keywords cannot be used as an identifier name.

5. What do you mean by Boolean literals?

Boolean literals are used to represent one of the Boolean values(True or false). Internally true

has value 1 and false has value 0.

6. What is Bitwise Operators? What are the types.

Bitwise operators work on each bit of data and perform bit-by-bit operation. In C++, there are three kinds of bitwise operators, which are:

- (i) Logical bitwise operators
- (ii) Bitwise shift operators
- (iii) Compliment operators

7. What is Bitwise one's compliment operator?

The bitwise One's compliment operator ~(**Tilde**), inverts all the bits in a binary pattern, that is, all 1's become 0 and all 0's become 1. This is a unary operator.

Example:

If **a** =15; Equivalent binary values of a

is **0000 1111**

Operator	Operation	Result								
		a	0	0	0	0	1	1	1	1
~	(~a)	(~a)	1	1	1	1	0	0	0	0
		$(\sim a) = 1111\ 0000_2 = -16_{10}$								

8. Define: Input operator.

C++ provides the operator >> to get input. It extracts the value through the keyboard and assigns it to the variable on its right; hence, it is called as **"Stream extraction"** or **"get from"** operator.

9. Define: Output Operator.

C++ provides << operator to perform output operation. The operator << is called the **"Stream insertion"** or **"put to"** operator. It is used to send the strings or values of the variables on its right to the object on its left. << is a binary operator.

10. What is cascading of operator?

To send more than one value at a time, << operator should be used for each constant / variable / expression. This is called **cascading of operator**.

11. Define: Cascading of I/O operators

The multiple use of input and output operators such as >> and << in a single statement is known as cascading of I/O operators.



12. What are the advantages of using float data types?

- > They can represent values between the integers.
- > They can represent a much greater range of values.

13. What are the four modifiers used in C++.

- (1) signed
- (2) unsigned
- (3) long
- (4) short

14. What is Garbage?

Declare a variable without any initial value, the memory space allocated to that variable will be occupied with some unknown value. These unknown values are called as **"Junk"** or **"Garbage"** values.

17

Extra 5 Marks

1. Explain : Lexical Units (Tokens)

The smallest individual unit in a program is known as a Token or a Lexical unit.

C++ has the following tokens:

- > Keyword
- Identifiers
- Literals (Constants)
- > Operators
- Punctuators

i) Keyword

- ➢ Keywords are the reserved words which convey specific meaning to the C++ compiler. They are the essential elements to construct C++ programs.
- \triangleright C++ is a case sensitive programming language so, all the keywords must be in lowercase.

Ex: int, void, break, do, if etc..

ii) Identifiers

Identifiers are the user-defined names given to different parts of the C++ program viz. variables, functions, arrays, classes etc.,

Rules for naming an identifier:

- The first character of an identifier must be an alphabet or an underscore (_).
- Only alphabets, digits and underscore are permitted. Other special characters are not allowed as part of an identifier.
- C++ is case sensitive as it treats upper and lower-case characters differently.
- Reserved words or keywords cannot be used as an identifier name.

iii) Literals (Constants)

L iterals are data items whose values do not change during the execution of a program. Therefore Literals are called as Constants.

C++ has several kinds of literals:

- Numeric constants
- Boolean constants
- Character constants
- String constants
- iv) Operators

The symbols which are used to do some mathematical or logical operations are called as **"Operators".**

In C++, The operators are classified on the basis of the number of operands.

- (i) Unary Operators Require only one operand
- (ii) **Binary Operators** Require two operands

Identifiers		
Nı	ım	
N	UM	
_a	dd 🖉	
tot	tal_sales	
tar	nilMark	

Numeric Constants

Boolean Constants

String Literals

Character Constants

Integer constants

Real Constants

(iii) Ternary Operators - Require three operands

v) Punctuators

Punctuators are symbols, which are used as delimiters, while constructing a C++ program. They are also called as **"Separators"**.

Ex:

```
Curly braces { }
Parenthesis ()
Square brackets [ ]
Comma ,
Semicolon ;
Colon :
Comments // /* */
```

2. Define: Literals (Constants) .Explain its Types with Example.

Literals are data items whose values do not change during the execution of a program. Therefore Literals are called as Constants.

C++ has several kinds of literals:

- i. Numeric constants
- ii. Boolean constants
- iii. Character constants
- iv. String constants

i) <u>Numeric Constants:</u>

As the name indicates, the numeric constants are numeric values, which are used as constants. Numeric constants are further classified as:

1. Integer Constants (or) Fixed point constants.

2. Real constants (or) Floating point constants.

(1) Integer Constants (or) Fixed point constants

Integers are whole numbers without any fractions. An integer constant must have at least one digit without a decimal point.

There are three types of integer constants:

(i)	Decimal	Ex: 785, -26
(ii)	Octal	Ex: 046, 027
(iii)	Hexadecimal	Ex: 0x123, 0A23

(2) Real Constants (or) Floating point constants

A real or floating point constant is a numeric constant having a fractional component.

Ex: 12.23

ii) <u>Boolean Literals</u>

Boolean literals are used to represent one of the Boolean values(True or false). Internally true has value 1 and false has value 0.

iii) <u>Character constant</u>

A character constant is any valid single character enclosed within single quotes. A character constant in C++ must contain one character and must be enclosed within a single quote.

Valid character constants : 'A', '2', '\$'

Invalid character constants : "A"

iv) <u>String Literals</u>

Sequence of characters enclosed within double quotes are called as String literals. By default, string literals are automatically added with a special character '\0' (Null) at the end.

Therefore, the string "welcome" will actually be represented as "welcome0" in memory and the size of this string is not 7 but 8 characters i.e., inclusive of the last character 0.

Valid string Literals : "A", "Welcome" "1234"

Invalid String Literals : 'Welcome', '1234'

3. What is data type? What are the types available in c++?

Data types are very essential elements to write even the most elementary programs. C++ provides a predefined set of data types for handling the data items.

In C++, the data types are classified as three main categories

(1) Fundamental data types

(2) User-defined data types and

(3) Derived data types.



- > The variables are the named memory locations to hold values of specific data types.
- In C++, the variables should be declared explicitly with their data types before they are actually used.

Syntax for declaring a variable:

<data type> <variable name>;

Example:

int num1;

4. Explain the fundamental data types in C++?

Fundamental (atomic) data types are predefined data types available with C++. There are five fundamental data types in C++:

- char
- int
- float
- double
- void.

Actually, these are the keywords for defining the data types.

(1) int data type:

Integers are whole numbers without any fraction. Integers can be positive or negative. Integer data type accepts and returns only integer numbers. If a variable is declared as an **int**, C++ compiler allows storing only integer values.

Ex: int x=10;

(2) char data type:

Character data type accepts and returns all valid ASCII characters. Character data type is often said to be an integer type, since all the characters are represented in memory by their associated **ASCII Codes**



(3) float data type:

If a variable is declared as float, all values will be stored as floating point values.

There are two advantages of using float data types.

(1) They can represent values between the integers.

(2) They can represent a much greater range of values.

Ex: float a=12.25;

(4) double data type:

This is for double precision floating point numbers. (precision means significant numbers after decimal point). The double is also used for handling floating point numbers. But, this type occupies double the space than float type. This means, more fractions can be accommodated in double than in float type. The double is larger and slower than type float.

Ex: double b=12543.21364;

(5) void data type:

The literal meaning for void is 'empty space'. Here, in C++, the void data type specifies an empty set of values. It is used as a return type for functions that do not return any value.

5. What is Expression? Explain.

- An expression is a combination of operators, constants and variables arranged as per the rules of C++. It may also include function calls which return values. (Functions will be learnt in upcoming chapters).
- An expression may consist of one or more operands, and zero or more operators to produce a value.

In C++, there are seven types of expressions, and they are:

- (i) Constant Expression
- (ii) Integer Expression
- (iii) Floating Expression
- (iv) Relational Expression
- (v) Logical Expression
- (vi) Bitwise Expression
- (vii) Pointer Expression

SN	Expression	Description	Example	
1	Constant Expression	Constant expression consist only constant values	int num=100;	
2	Integer Expression	The combination of integer and character values and/or variables with simple arithmetic operators to produce integer results.	sum=num1+num2; avg=sum/5;	
ň	Float Expression	The combination of floating point values and/or variables with simple arithmetic operators to produce floating point results.	Area=3.14*r*r;	Ne
4	Relational Expression	The combination of values and/or variables with relational operators to produce bool(true means 1 or false means 0) values as results.	x>y; a+b==c+d;	
5	Logical Expression	The combination of values and/or variables with Logical operators to produce bool values as results.	(a>b)&& (c==10);	
6	Bitwise Expression	The combination of values and/or variables with Bitwise operators.	x>>3; a<<2;	
7	Pointer Expression	A Pointer is a variable that holds a memory address. Pointer declaration statements.	int *ptr;	

6. Explain the Type Conversion.

The process of converting one fundamental type into another is called as "Type Conversion". C++ provides two types of conversions.

(1) Implicit type conversion(2) Explicit type conversion.

(1) Implicit type conversion:

An Implicit type conversion is a conversion performed by the compiler automatically. So, implicit conversion is also called as **"Automatic conversion"**.

This type of conversion is applied usually whenever different data types are intermixed in an expression. If the type of the operands differ, the compiler converts one of them to match with the other, using the rule that the "smaller" type is converted to the "wider" type, which is called as **"Type Promotion".**

For example:

```
#include <iostream>
using namespace std;
int main()
{
    int a=6;
float b=3.14;
    cout << a+b;
}</pre>
```

(2) Explicit type conversion

C++ allows explicit conversion of variables or expressions from one data type to another specific data type by the programmer. It is called as **"type casting"**.

Syntax:

(type-name) expression;

Example:

#include <iostream>
using namespace std;

int main()

float varf=78.685; cout << (int) varf;</pre>

In the above program, variable **varf is** declared as a **float** with an initial value 78.685. The value of **varf** is explicitly converted to an **int** type in cout statement. Thus, the final output will be 78.

M.VIJAYA KUMAR, MCA., M.Phil., B.Ed., PGDCA., PGT-COMPUTER TEACHER,

V.SUJITHA, MCA.,M.Phil.,

PGT-COMPUTER TEACHER,

SASTRA MATRICULATION HIGHER SECONDARY SCHOOL, KILPENNATHUR, TIRUVANNAMALAI 604601

Mobile No : 9655826843

Email : vijay28soft@gmail.com



ALL THE BEST

