Introduction to C++

One marks

1. Define token
   The tokens are smallest individual unit of program
2. Mention any two escape sequence

\n	New line
\t	Tab

3. What are binary operator
   These are operator requires 2 operand.
4. What is keyword?
   The key words are the identifiers or variables with pre assigned or reserved meaning. These names cannot be used as variable name in the program
5. Write the header file that holds setw() and endl
   #include<iomanip.h>
6. What are unary operator
   The unary operator requires one operand
7. What is an expression
   The expression is the valid combination of operator, variable and constant.
8. What is an identifier
   Identifier refers to the name of the variables
9. Who developed c++ Program
   Bjarne Stroustrup developed C++
10. What is comment?
    The non-executable part of the program
11. What is an expression?
    The expression is the valid combination of operator, variable and constant.
    E.g. A=b+7;

Two marks

1. **Mention any two characteristics of C++**
   Portability: we can carry program from place to other place and can run without much modification
   Brevity: The code size is short compare to other language
   Modular programming: The several source code compiled separately and then linked together
   Speed: Due to reduced size faster execution
2. **Mention any four token available in c++**
   Key words
   Identifiers
   Constants
   Punctuators
   Operators
   Character set

3. **What is iteration? Name the iteration construct**
   It is the ability of the programming language to execute certain set of statement again and again until the required condition is satisfied
   Example: while , for, do-while

4. **Mention any two string.h function**

strlen(str)	Gives the number of character in a given string str
strrev(str)	To convert the string to its reverse
strupr(str)	Converts the str to upper case
strlwr(str)	Converts the str to lower case

5. **Mention any two math.h function OR Write any two mathematical library function**

fabs(x)	Absolute value of real number x
abs(x)	Absolute value of integer number
pow(x,y)	The x to the power y
sqrt(x)	Square root of given number

6. **Mention any two function of header file ctype.h**

isalpha(ch)	It returns true if ch is alphabet (upper case or lower case) otherwise return false
isdigit(ch)	It returns true if ch is digit(0 to 9) otherwise return false
isalnum(ch)	It returns true if ch is alphabet or a digit otherwise return false
islower(ch)	It returns true if ch is lower case else return false

7. **What is the use of isalpha() and is digit()**

   Isalpha(): It returns true if ch is alphabet (upper case or lower case) otherwise return false
   isdigit():It returns true if ch is digit(0 to 9) otherwise return false

Three marks

1. **Summarize the rules for naming identifiers**
   The variable name should start with alphabet and it may contain anything
   Variable name should not contain any special character except underscore
   Variable name cannot start with digit
   The upper case and lower case are separate
   Reserved word cannot be used as variable name
   Usually 32 bit is the maximum number of character
2. **What is type conversion? Explain anyone with an example**
   Converting an expression from one data type to another data type.
   Explicit conversion or Type Casting: It is the process of converting form one data type to another data type by forcibly
   Syntax : **Variable-name = (new-data-type) variable-name;**
   Example

int a=2,b=3;
float avg;
avg=(float)a+b/2;

Implicit conversion :If the operand of 2 different type appears in the single expression then the lower type variable is converted in to higher type variable automatically

Example: int a;
char ch='A';
a=ch;
cout<<a; //automatically convert from character type to integer will gives output 65
i.e. it will print ASCII value of given character

(Explain any one type either implicit or explicit)

3. **What are the logical operator? Explain any two.**

These are the operator which will return the value true or false based on the condition .C++ has following three logical operators. Let us assume A=1 and B=0

Operator	Meaning	Example
&&	Logical AND operator. If both the operands are true (1) it gives true value otherwise false(0)	A&&B is false
||	Logical OR operator. If both the operands are false (0) it gives false value otherwise gives true(1)	A||B is true
!	Logical NOT operator. It is used to reverse the current state of the operand	!(A&&B) is true

4. **Explain ternary operator with example**

Ternary operator is also called as conditional, it requires 3 operand. The operator? and: are used to construct conditional expression

Syntax : **Variable= exp1? exp2:exp3;**

Where exp1, exp2 & exp3 are expressions.

exp1 is evaluated first, if it is non-zero(true), then the expression exp2 is evaluated and assigned to variable . If exp1 is false then the expression exp2 is evaluated and assigned to variable.

For example,
a=10;
b=15;
x=(a>b)?a:b;
Here, x is assigned the value of b.

Five marks

1. **Explain the structure of C++ program with an example**

The general structure is

Comment or documentation section
Linker section
Global variable declaration section
void main()
{
declaration section

executable statement section
}
user defined function

Consider the following program

```
1  #include<iostream.h>
2  void main(void)
 3          {
4            cout<<"Hello welcome to C++\n";
5            }
```

Annotation

1 This line uses preprocessor directive #include to include the contents of the header file iostream.h in the program. The iostream.h is a standard C++ header file and contains definition for input and output.
2 This line define the function called main. A function may have zero or more parameters, these always appear after the function name between a pair of brackets. The word void appearing between the brackets indicates that main as no parameters. The function may also have the return type. This always appear before the function name. All C++ function have one main function and the execution of program will begin from this point
3 This brace makes the beginning of the body of main
4 This line is a statement. A statement is a calculation step which may produce a value. The end of the statement is always marked with a semicolon (;). The information given inside the double quote (" ") will be printed as it is . The last character in the string (\n) is a new line character . The cout is a output stream object used to print the value on the screen. The symbol << is an output operator which takes an output stream as its left operand and expression is its write operand
5 This brace is used to end the body of main
The { and } are used to declare a set of statement together and the content present within side of this brace is called block.

2. Explain the characteristics of C++ in detail

Object-oriented programming: The possibility to orient programming to objects allows the programmer to design applications from a point of view more like a communication between objects rather than on a structured sequence of code. In addition it allows a greater reusability of code in a more logical and productive way.

Portability: we can carry program from place to other place and can run without much modification

Brevity: The code size is short compare to other language

Modular programming: The several source code compiled separately and then linked together

Speed: Due to reduced size faster execution

Machine independent: It will not depend on internal architecture of machine

Flexibility: The modification can be added easily

Wide range of library functions: By using library function we can reduce the line of code

System software development: Using this language we can develop system software, like editor, OS.

C compatibility: The code written in C can directly use in this

3. **What are binary operator Explain various binary operator with example**

   The binary operators are those operators that operate on two operands. They are as arithmetic, relational, logical, bitwise, and assignment operators.

   Arithmetic Operators C++ provides all the basic arithmetic operations such as addition, subtraction, multiplication, division and modulus operator

   Example: A+B

   Relational operators We often compare two quantities and depending on their relation take certain decisions. The relational operator always return true or false for example, 10 < 20 is true.

   Logical Operators used to combine two or more than two relational expression

   If a=10 and b=5 then, ((a==10) && (b>5)) returns false

   Assignment operators: Assignment operators are used to assign the result of an expression at right side to a variable at left side. We know usual assignment operator, **i.e. '='. Syntax:** Variable op=expression; Example x=10;

   Bitwise Operators: C++ has special operators known as bitwise operators for manipulation of data at bit level. These operators are used for testing the bits, or shifting them right or left. Bitwise operators may not be applied for float or double.

   Example: Bitwise shift operator

If X	0100 1001 1100 1011
X<<1	1001 0011 1001 0110

4. **Explain the constant with suitable example**

   It refers the fixed value and is not changes during the execution of the program. This constant can again divided into different type based on the type of the value it will store and are integer, character, floating point, and string

a) Integer constant: These are the constant without any fractional part and it is categorized as

   1) Decimal Constant: These are the constant starts with numbers ranging from 0 to 9. These numbers will not start with zero.

   2) Octal Constant: These are the constant starts with numbers ranging from 0 to 7. These numbers will starts with zero.

   3) Hexadecimal constant: These are the constant starts with numbers ranging from 0 to 9 and A to F. These numbers will starts with 0X. Here we can use upper case x or lowercase x along with zero. We can also use upper case A to F or we can use lower case a to f. In generally case does no matter.

b) Floating point constant: These constant contain the fractional value it means it contain decimal point and exponent. Using this we can represent the smaller or bigger number also.

   The exponent value is represented by e or E with optional +or – symbol. The E specifies power of ten

c) Character constant: A single character is enclosed in side of a pair of single quotation mark is called character constant.

d) String constant: A group of character enclosed inside of a pair of double quotation mark is called string constant. By default the compiler adds the null character ('\0') to the end of the string.

```
Example:
123        // integer constant
```

```
0123                    // octal constant
0X123                   // hexadecimal constant
12.3      //floating point constant
6.23E-6     // floating constant
'c'       // character  constant
"SMS"   // string constant
```

5. Explain the relational operator with example

Relational operators: We often compare two quantities and depending on their relation take certain decisions. For example, we may compare price of two items or age of two persons and so on. These comparisons can be done with the help of relational operators. An expression such as a < b or 1 < 20 containing a relational operator is termed as a relational expression. The value of relational expression is either one or zero. It is one if the specified relation is true and zero if the relation is false.

For example,

10 < 20 is true

But 20 <10 is false.

C ++ supports 6 relational operators.

Operator	**Meaning**	**Example**	**RESULT**
<	is less than	4.5 < -10	FALSE
< =	is less than or equal to	4.5 < = 10	TRUE
>	is greater than	5>2	TRUE
> =	is greater than or equal to	5>=5	TRUE
= =	is equal to	5==5	TRUE
! =	is not equal to	5!=5	FALSE

Relational expressions are used in decision statements such as, if and while to decide the course of action of a running program.

6. Explain five character built in function

Function	Meaning
isalpha(ch)	It returns true if ch is alphabet (upper case or lower case) otherwise return false
isdigit(ch)	It returns true if ch is digit(0 to 9) otherwise return false
isalnum(ch)	It returns true if ch is alphabet or a digit otherwise return false
islower(ch)	It returns true if ch is lower case else return false
isupper(ch)	It returns true if ch is upper case else return false
toupper(ch)	Converts ch from lower case to upper case
tolower(ch)	Converts ch from upper case to lower case

7. Explain with syntax five string function

Function	Meaning
strlen(str)	Gives the number of character in a given string str
strrev(str)	To convert the string to its reverse
strupr(str)	Converts the str to upper case
strlwr(str)	Converts the str to lower case
strcpy(str1,str2)	Copies the content of string to to string1
strcmp(str1,str2)	Compares the str1 and str2 and we get 3 possible output 0 if both string are equal 1 if first string is greater than second -1 if the first string is less than second This compression is case sensitive and comparison will takes place based on ASCII value
strcmpi(str1,str2)	Compares the str1 with str2 by ignoring case
strcat(str1,str2)	Content of str2 is added to the end of str1

8. Write the mathematical expressions into C++ expressions:

1. $A = \pi r^2$
2. s=(a^2+b^2+c^2)/4
3.area=$\sqrt{s(s-a)(s-b)(s-c)}$
4.volume=πr^2h
5.volume=3/4πr^3h

Ans:

1. A=3.14 *r*r;
2. S=(a*a +b*b +c*c)/4;
3. Area = sqrt (s*(s-a)*(s-b)*(s-c));
4. Volume = 3.14*r*r*h ;
5. Volume=3.0/4*3.14*r*r*r*h ;

1. **What is modular programming? Mention its advantages (**problem solving techniques**)**

The process of splitting or decomposing the complex programs into number of smaller units or sub program (modules) is called modularization and programming with such an approach is called ***modular programming***.

Advantages of modular programming:

Code Reusability

Debugging is easier

Building library

Portability

2. **Explain while construct with an programming example**(control statement)

While is a **pre-tested loop** structure. This structure checks the condition at the beginning of the structure. If condition is satisfied the set of statements are executed again and again until the condition is true or satisfied. When the condition becomes false, control is transferred out of the structure.

Example:

```
n = 10;
While (n > 0)
{
cout<<n<<"\t";
- - n;
}
cout<<"End of while loop \n";
```

Output: 10 9 8 7 6 5 4 3 2 1 End of while loop