C++ TOKENS,OPERATORS,CONDITION,LOOPS

C++ Tokens – smallest individual unit in a program

5 types of tokens are:

- Keyword(system defined names)
- Identifier(user defined names)
- Literals
    - Integer literals
    - Floating literal
    - Character literal- enclosed in single quotes. Contain only one character.
    - String literal-enclosed in double quotes. Contain more than one character
    - Escape Sequences or Backslash Character Constants - some nonprintable characters, as well as backslash (\ ) which can be expressed as escape sequences.

An escape sequence always starts with backslash followed by one or more special characters.

For example, a new line character is represented "\n" or endl

These are used in formatting output screen

- Punctuators—{},[],(),; ,, * >>, <<
- Operators-
    - 1. Arithmetic operators -- + ,- ,*, /, %,
    - 2. Relational operators >, <, >=, <=, !=, = =
    - 3. Logical operators && (and), ||(or) , ! (not)
    - 4. Assignment operators +=, -=, *=, /=, = , %=
    - 5. Increment or Decrement operators ++, --
    - 6. Conditional operator ? :
    - 7. Bit wise operators << , >>, ^, ~
    - 8. unary operator + , -, ++, --
    - 9. Special operators - sizeof(), comma (,)

1.Scope Resolution operator:
Scope:-Visibility or availability of a variable in a program is called as scope.
There are two types of scope.
i)Local scope
ii)Global scope

Local scope: visibility of a variable is local to the function in which it is declared.

Global scope: visibility of a variable to all functions of a program Scope resolution operator in "::" .
This is used to access global variables if same variables are declared as local

---

DATA TYPES:

A data type is used to indicate the type of data value stored in a variable.

Different types of data types are:

1.Primary (fundamental) data types – these are atomic data types

- int- store whole numbers. range of integer values is - 32,768 to +32,767. Occupies 2 bytes
- float- store number with decimal. 6 digits precision occupies 4 bytes of memory
- char- store characters. Occupies one byte
- double-Double floating point data type occupies 8 bytes of memory giving 14 digits of precision.
- void-used with function. void type has no values.
- bool-Boolean or logical data type having two values (usually denoted true and false).

2.Derived data types- these are derived from fundamental data types

- Array
- Pointer
- reference

3.User-defined data types- created by the user

- structure
- union
- class
- enumeration

Control statements:-The flow of execution of statements in a program is called as control.

Control statement is a statement which controls flow of execution of the program.

Control statements are classified into following categories.

1.Sequential control statements - Sequential control statements ensures that the instructions(or statements) are executed in the same order in which they appear in the program. i.e. By default system executes the statements in the program in sequential order.

2.Conditional control statements –Statements that are executed when a condition is true.

These statements are divided into two categories.

1.Decision making statements - These statements are used to control the flow of execution of a program by making a decision depending on a condition, hence they are named as decision

making statements. Decision making statements are of four types

1.Simple if

2.if else

3.nested if else

4.If else ladder

2.Switch case control statement – used for condition values. Only integer and character values it can accept

3.Loop control statements or repetitions

Loop statements can be divided into three categories as given below

1.while loop statement – entry controlled loop

2.do while loop statement – at least executed once. Exit controlled loop

3.for loop statement- entry controlled loop

 # include Directive

The # include directive instructs the compiler to read and include another file in the current file. The compiler compiles the entire code.
A header file may be included in one of two ways.
include <iostream.h> or include "iostream.h"

The header file in angle brackets means that file reside in standard include directory.
The header file in double quotes means that file reside in current directory.

1. Why main function is so special in C++. Give at least two reasons.

Ans: Whenever a C++ program is executed only the main function is executed. The execution of the program starts and end at main().

2 What is the purpose of a header file in a program?

Ans: The c++ language contains some statement only and not any built-in function. Rather it provides standard library that contains files storing the standard functions these files are know as header files. Header files provide function prototype , definition of library function.

C++ Tokens:

The smallest individual unit of the program is known as Token.

Keywords	Words reserved by the language with a special meaning. Like int, float, switch, case etc
Identifiers	Long sequence of letters & digits to identify a memory block, program unit or program objects. Like name, a, x, A, X, date, file1, file2 etc.

Literals Constants that never changes their value during the execution of a program.
Type : Integer Constant (Decimal ,Octal (preceded by 0) Hexadecimal – (preceded by 0x or 0X)
• Character Constant („z", „A", „\a", „\t", „\"", „0", „3")
• Floating Constant (-13.0, -0.00065, 1.52E07, 0.172E-3)
• String Constants ("abc\0", "Computer Science\0")

Punctuators [ ] ( ) { } , ; : * = #
Operators <<, >>, +, -, *, /, % , ++, --, ==, <, >, <=, >=, !=, &&, | |, !

1. Write two advantages of using #include complier directive.
   Ans: The preprocessor directive # include tells the compiler to insert another file into the source file. The two advantages are;
1. The program is broken down into modules thus making it more simplified.
2. More library functions can be used at the same time size of the program is retained.

#define directive (Defined constants)

We can define our own names for constants that we use very often without having to resort to memory-consuming variables, simply by using the #define preprocessor directive. Its format is:
`#define identifier value`
For example:

```
#include<iostream.h>
#define PI 3.14159 // This defines two new
constants: PI and NEWLINE. #define NEWLINE '\n' // Once they are defined,
they can be used
      // in the code in place of their values
int main ()
{
double r=5.0; // radius
double circle;
circle = 2 * PI * r;
cout << circle;
cout << NEWLINE;
return 0;
}
```

Output :

```
31.4159
```

1. Illustrate the use of #define in C++ to define a macro.
   Ans #define preprocessor directive is used to define symbolic constant in a program. The #define identifier replacement is also known as a macro definition or simply a macro macros may be define with zero or more arguments. A macro without argument is used to define a symbolic constant and a macro with arguments is used to replace expressions or code fragments in a program. A macro define is not terminated by semicolon. Example:

```
#define LIMIT 10 //Symbolic constant
#define sqr(x) (x*x) //Macro
```

2. What is the difference between macros and function?
   Ans With Macros, a program runs faster as compared to functions. However, macros increase the program size a macros statement is replaced with its expansion wherever it is called. Whereas, functions occupy less memory space as the function call statement passes the program control to the function instead of replacing the call with the function body.

Declared constants (const)

With the const prefix we can declare constants with a specific type in the same way as we would do with a variable. For example:

```
const int pathwidth = 100;
const char tabulator = '\t';
```

Here, pathwidth and tabulator are two typed constants. They are treated just like regular variables except that their values cannot be modified after their definition.

Type Conversion in C++:

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

1. Implicit conversion (Type promotion): Implicit conversions do not require any operator. They are automatically performed when a value is copied to a compatible type.

2. Explicit type conversion (Type Casting): Type casting operators allow us to convert adatum of a given type to another. There are several ways to do this in C++. The simplest one, which has been inherited from the C language, is to precede the expression to be converted by the new type enclosed between parentheses (()):

```
int i;
float f = 3.14;
i = (int) f;
```

The previous code converts the float number 3.14 to an integer value (3), the remainder is lost. Here, the typecasting operator was (int).

Question : What is compiler and linker?
Answer : Compiler - It is a program which converts the program written in a programming language to a program in machine language. Linker - It is a program which links a complied program to the necessary library routines, to make it an executable program.

Question : Why is char often treated as integer data type in C++ ?
Answer : The memory implementation of char data type is in terms of the number code. Therefore, it is said to be another integer data type.

Question : What is type conversation in C++ ?

Answer : When two operands of different data types are encountered in the same expression, the variable of lower data type is automatically converted to the data tpes of variable with higher data type, and then the expression is calculated.
For example: int a=98;
float b=5;
cout << a/3.0; //converts to float type,
since 3.0 is of float type.
cout << a/b; // converts a temporarily to float type, since b is of float type, and gives the result 19.6.

Question : What is type casting in C++ ?
Answer : Type casting refers to the data type conversions specified by the programmer, as opposed to the automatic type conversions. This can be done when the compiler does not do the conversions automatically. Type casting can be done to higher or lower data type.
For example :
cout << (float)12/5;
//displays 2.4, since 12 is converted to float type.

Question : What is the effect of absence of break in switch case statement in C++ ?
Answer :
The break keyword causes the entire switch statement to exit, and the control is passed to statement following the switch.. case construct. Without break, the control passes to the statements for the next case. The break statement is optional in switch..case construct.

Question : In a control structure switch-case what is the purpose of default in C++ ?
Answer : This keyword gives the switch…case construct a way to take an action if the value of the switch variable does not match with any of the case constants. No break statement is necessary after default case, since the control is already at the end of switch..case construct. The default is optional in case of switch…case construct.

Question : What is the difference between while and do-while loop in C++ ?

Answer : While is an Entry Controlled Loop, the body of the loop may not execute even once if the test expression evaluates to be false the first time, whereas in do..while, the loop is executed at least once whether the condition holds true the first time or not.

Question : What is the difference between call by value and call by reference in a user defined function in C++?
Answer : The value of the actual parameters in the calling function do not get affected when the arguments are passed using call by value method, since actual and formal parameters have different memory locations. The values of the formal parameters affect the values of actual parameters in the calling function, when the arguments are passed using call by reference method. This happens since the formal parameters are not allocated any memory, but they refer to the memory locations of their corresponding actual parameters

Question : What is preprocessor directive?

Answer : A preprocessor directive is an instruction to the complier itself. A part of compiler called preprocessor deals with these directives, before real compilation process. # is used as preprocessor directive in C++.

Question What is the difference between typecasting and automatic type conversion? Explain with suitable example.
Ans:**Type Casting**: It is an user define explicit type conversion in which operand in an expression is forced to be of a specific data type e.g. (type)expression

Automatic type conversion: In case of an expression in which operands different data type are mixed an implicit type conversion is performed by the compiler which automatically converts all operands upto the type of largest operands.

---

sizeof( )operator:

This operator accepts one parameter, which can be either a type or a variable itself and returns the size in bytes of that type or object:

```
a = sizeof (char);
```

This will assign the value 1 to „a" because char is a one-byte long type. The value returned by sizeof is a constant, so it is always determined before program execution.

typedef keyword:

Using the keyword typedef, we can create an alias (a synonym) for existing fundamental or compound datatypes in C++. **Syntax:**

```
typedef existing_type new_type_name ;
```

where existing_type is a C++ fundamental or compound type and new_type_name is the name for the new type we are defining. For example:

```
typedef char C;
typedef unsigned int WORD;
typedef char * pChar;
typedef char field [50];
```

typedef does not create different types. It only creates synonyms of existing types. That means that the type of `myword` can be considered to be either `WORD` or `unsigned int`, since both are in fact the same type.

---

1 Differentiate between a Logical Error and Syntax Error. Also give suitable examples of each in C++.

Ans Logical Error. A logical error is that error which causes a program to produce incorrect or undesired output.

An incorrectly implemented algorithm or use of a variable before its initialization, or unmarked end for a loop, or wrong parameters passed are causes logical errors. These must be handled carefully.

For instance, if we are trying to print the table of a number 5 and if we say

```
counter=1;
while(counter>8)
{ cout<<n*counter;
counter=counter+1;
}
```

Here the loop would not be executed even once as the condition (counter>8) is not fulfilled at all. Therefore, no output will be produced. Such an error is logical error.

Syntax Error: Syntax errors occur when rules of a programming languages (syntax) is misused. Ie when a grammatical rule of C++ is violated.

Eg (i) c=a+b

In this statement, since there is no semicolon at the end of the statement, there will occurs a syntax error.

(ii)cin<<a; In this statement, since stream insertion operator (<<) has given instead of stream extraction operation(>>), there will occurs a syntax error.

2 What do you understand by Syntax Error, logical Error & Run time error.

Ans: **Syntax Error**: Those error that comes at the time of compilation are known as syntax error like as statement terminator missing.

Logical Error: Logical errors occur due to wrong analysis of the program. Due to logical error program does not give the desired result.

Runtime Error: Those errors that comes during execution are known as run-time error. Example:

1. Division by zero
2. Stack over flow
3. Array index out of bound
