

CS 201

FINAL TERM PREPARATION

IMP MCQS + SHORT NOTES

BY

VU LEARNING ZONE

VU
LEARNING ZONE

YOUTUBE VIDEO LINK

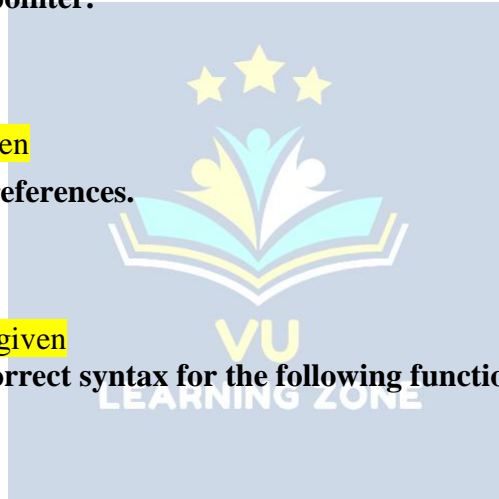
https://youtu.be/_HoulT-7xSI

WHATSAPP GROUP FOR FILES

<https://chat.whatsapp.com/JZ8MypU1DUdDLZME4psa>
[KL](#)

MCQS

1. For binary member operators, operands on the _____ drives (calls) the operation.
 1. Left
 2. Right
 3. Both left and right
 4. None of the given
2. We cannot increment _____.
 1. pointers
 2. arrays
 3. references
 4. variables
3. We can _____ pointer.
 1. increment
 2. decrement
 3. reassign
 4. all of the given
4. We can _____ references.
 1. increment
 2. decrement
 3. reassign
 4. None of the given
5. What will be the correct syntax for the following function call?
float add (int &);
 1. add(int x);
 2. add(&x);
 3. add(x);
 4. add(*x);
6. An instance of a class is called _____.
 1. structure
 2. data type
 3. object
 4. member function
7. The _____ is called automatically when an object destroys
 1. destructor
 2. constructor
 3. main program
 4. default constructor
8. The destructor is used to _____.
 1. allocate memory
 2. deallocate memory
 3. create objects
 4. allocate static memory



9. _____ data isn't accessible by non-member functions or outside classes.

1. Public
2. private
3. Static
4. Globally declared

10. Member functions of the class _____ main program.

1. are not accessible
2. are accessible from
3. are defined within the
4. are private to

11. Overloading means :

1. Using the same name to perform multiple tasks or different tasks depending on the situation.
2. Using the different name to perform multiple tasks or different tasks depending on the situation
3. Using the same name to perform multiple tasks or same tasks depending on the situation
4. Using the same name to perform difficult tasks or complex tasks and it does not depend on the situation

12. The main advantage of function overloading is _____.

1. The program becomes portable
2. The program becomes complex
3. The function becomes inline
4. The program becomes more readable

13. You cannot overload the _____ operator.

1. ? :
2. *
3. /
4. ++

14. In C++, a variable can be declared anywhere in the program this will increase _____.

1. writability
2. readability
3. portability
4. efficiency

15. Memory allocated from heap or free store _____.

1. can be returned back to the system automatically
2. can be allocated to classes only
3. cannot be returned back unless freed explicitly using malloc and realloc
4. cannot be returned back unless freed explicitly using free and delete operators

16. We cannot use _____ pointer for storing and reading data from it.

1. NULL
2. integer
3. double
4. zero

17. The dynamic memory allocation uses _____ whereas static memory allocation uses _____.

1. heap , stack
2. stack , lists
3. array , stack
4. classes , array

18. What will be the output of the given code?

```
#include #define MAX( A, B ) ((A) > (B) ? (A) : (B))
void main() {
int i, x, y;
x = 23;
y = 45;
i = MAX( x++, y++ );
// Side-effect: // larger value incremented twice
cout << "x = " << x << " y = " << y << '\n';
}
```

1. x=23 y=45
2. x=24 y=46
3. x=24 y=47
4. x=22 y=47

19. NULL has been defined in _____ header file.

1. Iostream.h
2. Stdlib.h
3. Stdio.h
4. String.h

20. Symbolic constant PI can be defined as:

1. #define PI 3.14 ;
2. #define PI 3.14
3. #define PI=3.14
4. # include pi= 3.14

21. The friend function of a class can have access _____.

1. to the public data members only
2. to the private data members
3. to the protected data members
4. to the main program

22. C++ was developed by _____.

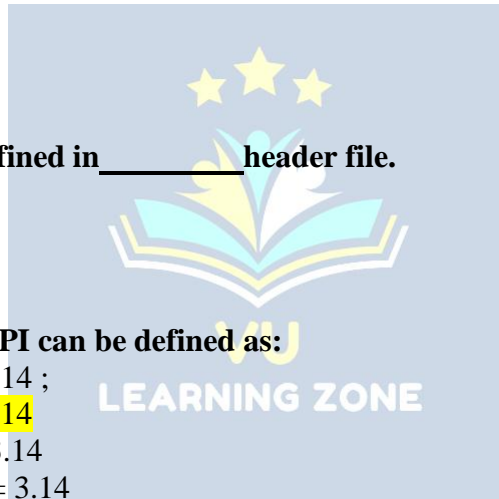
1. Charles Babbage
2. Graham Bell
3. Bejarne Stroustrup
4. Von Nuemann

23. Once the _____ are created, they exist for the life time of the program.

1. local variables
2. non static variables
3. static variables
4. automatic variables

24. Encapsulation means _____.

1. that the data of a class cannot be accessed from outside



2. that the data of a class can be accessed from outside
 3. the data becomes public
 4. that the data can be accessible anywhere within a main program
25. An address is a _____, while a pointer is a _____.
1. constant, variable
 2. variable, constant
 3. global, variable
 4. non static variable, constant
26. The syntax of declaration of a function that returns the reference to an integer is _____.
1. int & myfunc();
 2. int myfunc();
 3. int myfunc() &;
 4. integer & myfunc();
27. Which one of the following is mandatory preprocessor directive for c++?
1. #undef
 2. #include
 3. #undef
 4. All of the given
28. The members of a class declared with the keyword struct are _____ by default.
1. static
 2. Private
 3. protected
 4. public
29. getch() is a _____ function and defined in _____ header file.
1. user-define function , conio.h
 2. built-in function , conio.h
 3. built-in function, stlib.h
 4. built -in function, iostream.h
30. _____ Operators are the ones that require two operands on both sides of the operator.
1. Double
 2. Tow sided
 3. Binary
 4. None of the given
31. _____ will return the number of bytes reserved for a variable or data type.
1. sizeof operator
 2. free operator
 3. void pointer
 4. new operator
32. _____ are not available in C language.
1. User defined functions
 2. Built in functions
 3. Library functions
 4. Inline functions
33. The members of a class declared without any keyword are _____ by default.
1. protected
 2. private

3. public
4. constant

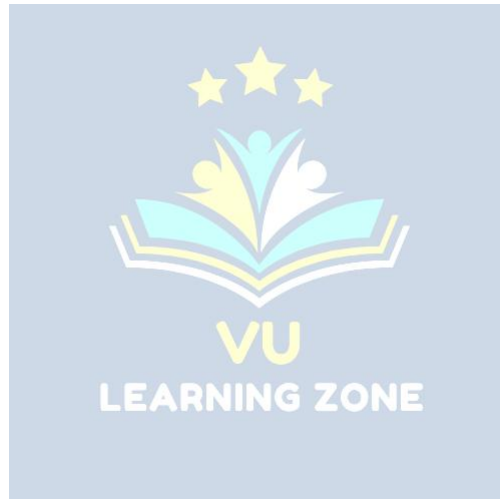
34. For console input and output we use _____.

1. conio.h header file
2. stdlib.h header file
3. process.h header file
4. getch.h header file

35. The name of the destructor is the same as that of a class proceeding with a _____.

1. & sign
2. # sign
3. @ sign
4. ~ sign

36. A reference cannot be NULL it has to point a data type.



1. True
 2. False
37. A pointer is_____.
1. the address of a variable
 2. an indication of the variable to be accessed next
 3. a variable for storing address
 4. the data type of an address variable
38. Constructor is a special function, called whenever we_____.
1. create a function
 2. instantiate an object of a class
 3. destroy an object
 4. create a class
39. Symbolic constant PI can be defined as:
1. #define PI 3.14;
 2. #define PI 3.14
 3. #define PI=3.14
 4. # include pi=3.14
40. Object code is machine code but it is not _____ and _____.
1. relocatable, executable
 2. faster, efficient
 3. compiled, debugged
 4. tested, compiled
41. The default visibility for the data members of the class is
1. private
 2. protected
 3. public
 4. accessible outside the class
42. The _____ is called automatically when an object destroys.
1. destructor
 2. constructor
 3. main program
 4. default constructor
43. Constructor is special type of function :
1. which has no return type
 2. which returns NULL pointer
 3. which returns zero
 4. which returns integer type data
44. _____ Variables are those that are defined outside of main.
1. Local
 2. Dynamic
 3. Global
 4. Static
45. Within the statement obj1=obj2; obj1 will call the assignment operator function and obj2 will be passed as an argument to function.
1. True
 2. False

46. When the compiler overload the assignment (=) operator by default then

1. Class members are not assigned properly
2. Compiler does not allow default assignment operator
3. Compiler does member wise assignment.
4. None of the given

47. It is possible to return an object from function using this pointer.

1. True
2. False

48. Overloaded assignment operator must be

1. Member function of class
2. Non-member function of class
3. Friend function of class
4. Global function

49. Let suppose

`int a, b, c, d, e;`

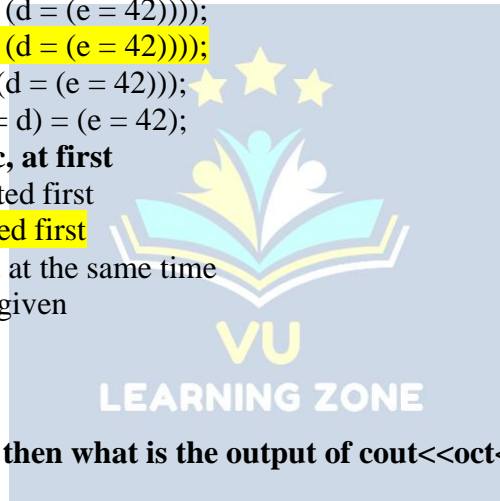
`a = b = c = d = e = 42;`

This can be interpreted by the compiler as

1. `(a = b = (c = (d = (e = 42))));`
2. `a = (b = (c = (d = (e = 42))));`
3. `a = b = (c = (d = (e = 42)));`
4. `(a = b) = (c = d) = (e = 42);`

50. In statement `a+b+c`, at first

1. `a+b` is executed first
2. `b+c` is executed first
3. All executed at the same time
4. None of the given



51. Suppose `int i = 10;` then what is the output of `cout<<oct<<i;`

1. 10
2. 11
3. 12
4. 13

52. `ostream` is a _____ operator.

1. dependent
2. member
3. standalone
4. None of the given

53. _____ must be included to use stream manipulation in your code.

1. `conio.h`
2. `iostream`
3. `stdlib.h`
4. `omanip`

54. _____ operators are the ones that require only one operator to work.

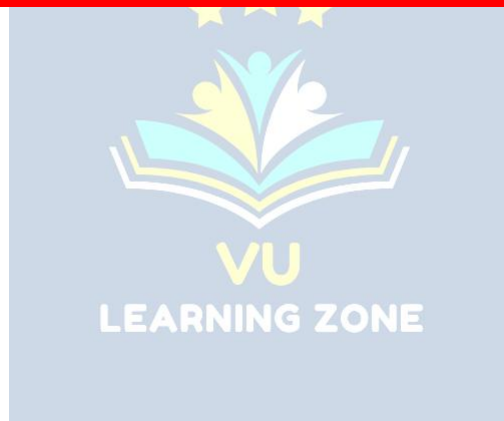
1. Unit

2. Unary
3. Single
4. None of the given
55. The endl and flush are _____.
1. Functions
 2. Operators
 3. Manipulators
 4. Objects
56. When operator function is implemented as member function then return type of function _____.
1. Must be an object of same class
 2. Must be user-defined data type
 3. Must be built-in data type
 4. Can be any data type
57. When a variable is defined as static in a class then _____.
1. Separate copy of this variable is created for each object
 2. Only one copy is created for all objects of this class
 3. A copy of this variable is created for only static objects
 4. None of the given
58. Automatic variables are created on _____.
1. Heap
 2. Free store
 3. Static storage
 4. stack
59. `cout << i << " ";`
`cout << d << " ";`
`cout << f;`
- Above statements can be written within statement of one line as:
1. `cout << i << " " << d " " << f << ;`
 2. `cout << i << << d << << f <<;`
 3. `cout << i << " " << d << " " << f;`
 4. `cout << i << " " << d << " " f <<;`
60. dec, hex, oct are all _____.
1. Member functions
 2. Objects of input/output streams
 3. Parameterized manipulators
 4. Non-parameterized manipulators
61. What will be the output of following statement?
`cout << setfill('0') << setw(7) << 128;`
1. 0128128
 2. 0000128
 3. 1280000

4. 0012800
62. Which of the following syntax is best used to delete an array of 5 objects named 'string' allocated using new operator.
1. delete string;
 2. delete []string;
 3. delete string[];
 4. delete string[5];
63. If we have a program that writes the output data(numbers) to the disc, and if we collect the output data and write it on the disc in one write operation instead of writing the numbers one by one.
In the above situation the area where we will gather the number is called
1. Heap
 2. Stack
 3. Buffer
 4. Cache
64. The first parameter of operator function for << operator _____.
1. Must be passed by value
 2. Must be passed by reference
 3. Can be passed by value or reference
 4. Must be object of class
65. The second parameter of operator function for >> operator must always be passed
1. By reference
 2. Function takes no argument
 3. By value
 4. None of the given
66. The only operator that the compiler overloads for user define data type by default is
1. Plus (+) operator
 2. Minus (-) operator
 3. Assignment (=) operator
 4. Equal (==) operator
67. Consider the following code, the printed value will be converted into:
- ```
int n=10;
cout <<oct<<n;
```
1. Base 8
  2. Base 2
  3. Base 10
  4. Decimal number system
68. \_\_\_\_\_ variables are defined in the main.
1. Global
  2. Dynamic
  3. Local
  4. All
69. ostream class is \_\_\_\_\_ and not under our control.
1. user-defined
  2. built-in
  3. both user-defined and built-in

**SUJECTIVE FILE**

**BY VU LEARNING ZONE**



## **What is Dev. C++?**

Bloodshed Dev-C++ is a full-featured Integrated Development Environment (IDE) for the C/C++ programming language. It uses Ming port of GCC (GNU Compiler Collection) as its compiler. Dev-C++ can also be used in combination with Cygwin or any other GCC based compiler.

## **Why Use Dev. C++ instead of C++ DOS IDE?**

Dev-C++ is a free IDE for Windows that uses either MinGW or TDMGCC as underlying compiler. Originally released by Bloodshed Software, but abandoned in 2006, it has recently been forked by Orwell, including a choice of more recent compilers.

## **What is #Include directive or Copy books or Header File?**

Many programming languages and other computer files have a directive, often called include (as well as copy and import), that causes the contents of a second file to be inserted into the original file. These included files are called copybooks or header files.

## **Why we use curly braces in c?**

Curly braces are used to group a set of statements. Often, we use them along with loops and conditional statements in order to avoid confusion and to define a clear scope.

**What is Main () function?** In C, the "main" function is treated the same as every function, it has a return type (and in

some cases accepts inputs via parameters). The only difference is that the main function is "called" by the operating system when the user runs the program.

### **How many data types are in C?**

Void, Int, Float, Character, String

### **How many types are for integers?**

Int or signed int unsigned int, short int or signed short int, long int or signed long int, unsigned long int

### **How many types are for Float data type?**

Float, double, long double

### **What is mean by using namespace std; in C?**

The built in C++ library routines are kept in the standard namespace. That includes stuff like cout, cin, string, vector, map, etc. Because these tools are used so commonly, it's popular to add "using namespace std" at the top of your source code so that you won't have to type the std: prefix constantly.

### **Define the memory size of int variable types?**

Type Size in Bytes Range

int or signed int 2 -32,768 to 32767

unsigned int 2 0 to 65535

short int or signed short int    1    -128 to 127 long int or  
signed long int 4    -2,147,483,648 to 2,147,483,647  
unsigned long int    4    0 to 4,294,967,295

### **Define the memory size of Float Variable types?**

| Type        | Size(bytes) | Range                  |
|-------------|-------------|------------------------|
| Float       | 4           | 3.4E-38 to 3.4E+38     |
| double      | 8           | 1.7E-308 to 1.7E+308   |
| long double | 10          | 3.4E-4932 to 1.1E+4932 |

### **How declares a variable?**

The Syntax of Declare variable is datatype variable name; where the datatype is the type of data which can hold a variable name might be anything except the C directives and reserve words.

### **What is namespace?**

Consider a situation, when we have two persons with the same name, Zara, in the same class. Whenever we need to differentiate them definitely, we would have to use some additional information along with their name, like either the area if they live in different area or their mother or father name, etc. Same situation can arise in your C++ applications. For example, you might be writing some code that has a function called xyz () and there is another library available which is also having same function xyz (). Now the compiler has no way of knowing which

version of xyz () function you are referring to within your code. A namespace is designed to overcome this difficulty and is used as additional information to differentiate similar functions, classes, variables etc. with the same name available in different libraries. Using namespace, you can define the context in which names are defined. In essence, a namespace defines a scope.

### **How define a namespace?**

A namespace definition begins with the keyword namespace followed by the namespace name as follows

```
namespace namespace_name {
 // code declarations
}
```

To call the namespace-enabled version of either function or variable, prepend the namespace name as follows:

```
name: code; // code could be variable or function.
```

### **What are reserve words in C?**

Reserve or Keywords are the words that C language use for his personal use a user cannot use reserve words as variable, function or class name. Example: IF, Else, Switch, For, While, Float etc.

### **What rules are for declaring a variable?**

1. Every variable name should start with alphabets or underscore (\_).

2. No spaces are allowed in variable declaration.
3. Except underscore (\_) no other special symbol are allowed in the middle of the variable declaration.
4. Maximum length of variable is 8 characters depend on compiler and operation system.
5. Every variable name always should exist in the left-hand side of assignment operator.
6. No keyword should access variable name.
18. What is difference between declaration and initialization of a Variable?

For a variable, a definition is a declaration which allocates storage for that variable.

Initialization is the specification of the initial value to be stored in an object, which is not necessarily the same as the first time you explicitly assign a value to it.

### **What is cin and cout?**

cout is used for output, cin for input. Cout and cin are not key words in the C++ language. They are variables, instances of classes that have been declared in <iostream>.

### **Define local and global variable?**

Variables that are declared inside a function or block are local variables. They can be used only by statements that are inside

that function or block of code. Local variables are not known to functions outside their own. Global variables are defined outside of all the functions, usually on top of the program. The global variables will hold their value throughout the life-time of your program. A global variable can be accessed by any function. That is, a global variable is available for use throughout your entire program after its declaration.

### **Define Syntax Error?**

A syntax error is an error in the syntax of a sequence of characters or tokens that is intended to be written in a particular programming language. For compiled languages, syntax errors are detected at compile-time. A program will not compile until all syntax errors are corrected.

### **What is Run-Time Error?**

An error that occurs during the execution of a program. In contrast, compile-time errors occur while a program is being compiled. Runtime errors indicate bugs in the program or problems that the designers had anticipated but could do nothing about. For example, running out of memory will often cause a runtime error.

### **What is Logical Error?**

A logic error is a bug in a program that causes it to operate incorrectly, but not to terminate abnormally (or crash). A logic error produces unintended or undesired output or other behavior, although it may not immediately be recognized as such.

## What are arithmetic operators?

C language has 5 arithmetic operators (+, -, \*, /, %). Arithmetic operators are used to perform arithmetic operations in c programming.

## Define Logical Operators?

C Language have 3 types of Logical operator or conditional operators.

| Operator | Description | Example |
|----------|-------------|---------|
|----------|-------------|---------|

|    |                                                                                                                     |
|----|---------------------------------------------------------------------------------------------------------------------|
| && | Called Logical AND operator. If both the operands are non-zero, then the condition becomes true. (A && B) is false. |
|----|---------------------------------------------------------------------------------------------------------------------|

|  |                                                                                                                       |
|--|-----------------------------------------------------------------------------------------------------------------------|
|  | Called Logical OR Operator. If any of the two operands is nonzero, then the condition becomes true. (A    B) is true. |
|--|-----------------------------------------------------------------------------------------------------------------------|

|   |                                                                                                                                                                               |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ! | Called Logical NOT Operator. It is used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false. !(A && B) is true. |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## What is Conditional operator?

A ternary operator or Conditional Operator is an operator that takes three arguments. The arguments and result can be of different types. Many programming languages that use C-like syntax feature a ternary operator, `?:`, which defines a conditional expression. Since this operator is often the only existing ternary operator in the language, it is sometimes simply referred to as "the ternary operator". In some languages, this operator is referred to as "the conditional operator". Another example for a ternary operator is `between`, as used in SQL and Python.

## Difference between operators and operands?

Operands are the objects that are manipulated and operators are the symbols that represent specific actions. For example, in the

expression.  $5 + x$ . and 5 are operands and + is an operator. All expressions have at least one operand.

### **Describe about file streams? Data Type Description**

of stream This data type represents the output file stream and is used to create files and to write information to files.

if stream This data type represents the input file stream and is used to read information from files.

stream This data type represents the file stream generally, and has the capabilities of both of stream and if stream which means it can create files, write information to files, and read information from files.

### **What are conditional statements?**

In computer science, conditional statements, conditional expressions and conditional constructs are features of a programming language, which perform different computations or actions depending on whether a programmer-specified Boolean condition evaluates to true or false. Example: IF. Then, Switch.

### **What are repetition statements?**

In C programming Language it is often necessary to repeat the same block of code a given number of times, or until a certain condition is met. Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. You can use one or more loops inside any other while, for, or do. While loop.

### **Define Nested If and Nested Loop?**

In C++ we can use if statement in the else block. Or we can also include if block in another if block, it's called nested IF. The placing of one loop inside the body of another loop is called nesting. When you "nest" two loops, the outer loop takes control of the number of complete repetitions of the inner loop. While all types of loops may be nested, the most commonly nested loops are for loops

### **What is pre-processor Directive?**

The C preprocessor modifies a source file before handing it over to the compiler, allowing conditional compilation with #ifdef, defining constants with #define, including header files with #include.

### **Why use strcpy () function?**

This function use to copy string from one string variable to other string variable. Example strcpy (s1, s2) copy string from s2 to s1.

### **What is a Function?**

A function is a group of statements that together perform a task. Every C program has at least one function, which is main (), and all the most trivial programs can define additional functions. You can divide up your code into separate functions.

### **What are parameters of a function?**

In computer programming, a parameter is a special kind of variable, used in a subroutine to refer to one of the pieces of data provided as input to the subroutine. These pieces of data are

called arguments. An ordered list of parameters is usually included in the definition of a subroutine, so that, each time the subroutine is called, its arguments for that call can be assigned to the corresponding parameters.

### **What is a pointer?**

The variable that stores the address of another variable is called a pointer. Pointers are a very powerful feature of the language that has many uses in lower-level programming.

### **Difference between a pointer variable and a Variable?**

Both are variables a pointer holds the memory address of a variable and a Variable that hold his value both variables must be same type to work with each other.

### **What is a Class?**

A class in C++ is a user defined type or data structure declared with keyword class that has data and functions (also called methods) as its members whose access is governed by the three access specifies private, protected or public (by default access to members of a class is private).

### **What is a Class object?**

Object is a class type variable. Objects are also called instance of the class. Each object contains all members (variables and functions) declared in the class. We can access any data member or member function from object of that class using. Operator.

## **How can we refer to the global variable if the local and the global variable names are same?**

We can apply scope resolution operator: for the scope of global variable. The scope resolution operator helps to identify and specify the context to which an identifier refers, particularly by specifying a namespace. The specific uses vary across different programming languages with the notions of scoping. In many languages the scope resolution operator is written “.”.

## **What is recursion?**

Recursion is the process of repeating items in a self-similar way. In programming languages, if a program allows you to call a function inside the same function, then it is called a recursive call of the function.

## **What is an identifier?**

An Identifier can only have alphanumeric characters (a-z, A-Z, 0-9) and underscore (\_). The first character of an identifier can only contain alphabet (a-z, A-Z) or underscore (\_). Identifiers are also case sensitive in C. For example, name and Name are two different identifiers in C.

## **What is friend class and friend Function?**

Friend Function: A C++ friend functions are special functions which can access the private members of a class. They are considered to be a loophole in the Object-Oriented Programming

concepts, but logical use of them can make them useful in certain cases.

**Friend Class:** A C++ friend class is a special class that can access the members, objects and methods of another class.

## **What are constructors and destructors?**

**Constructor:** A class constructor is a special member function of a class that is executed whenever we create new objects of that class. A constructor will have exact same name as the class and it does not have any return type at all, not even void.

Constructors can be very useful for setting initial values for certain member variables.

**Destructors:** A destructor is a special member function of a class that is executed whenever an object of its class goes out of scope or whenever the delete expression is applied to a pointer to the object of that class. A destructor will have exact same name as the class prefixed with a tilde (~) and it can neither return a value nor can it take any parameters. Destructor can be very useful for releasing resources before coming out of the program like closing files, releasing memories etc.

## **What is the difference between actual and formal parameters?**

The parameters sent to the function at calling end are called as actual parameters while at the Receiving of the function definition called as formal parameters.

## **What is difference between including the header file with-in angular braces <> and double quotes ""?**

If a header file is included with in <> then the compiler searches for the particular header file

Only with in the built in include path. If a header file is included with in

"", then the compiler. Searches for the particular header file first in the current working directory, if not found then in the built in include path

## **If a pointer is declared for a class, which operator can be used to access it?**

Arrow operator (->) is used for accessing members of structure using pointer variable, whenever we declare structure variable then member can be accessed using the dot operator. But when pointer to a structure is used then arrow operator is used. Both dot and arrow operator serve same function to access member of structure.

## **What is function overriding?**

Function overloading (also method overloading) is a programming concept that allows programmers to define two or more functions with the same name. Each function has a unique signature (or header), which is derived from: function/procedure name. Number of arguments. Arguments' type.

## **Is it possible to have Virtual Constructor? If yes, how? If not, why not possible?**

There is nothing like Virtual Constructor. The Constructor can't be virtual as the constructor

Is a code which is responsible for creating an instance of a class and it can't be delegated to any other object by virtual keyword mean.

### **What is a dangling pointer?**

A dangling pointer arises when you use the address of an object after its lifetime is over. This may occur in situations like returning addresses of the automatic variables from a function rousing the address of the memory block after it is freed.

### **What is a smart pointer?**

A smart pointer is a C++ class that mimics a regular pointer in syntax and some semantics, but it does more. Because smart pointers to different types of objects tend to have a lot of code in common, almost all good-quality smart pointers in existence are template by the pointe type.

### **What is the difference between a pointer and a reference?**

A reference must always refer to some object and, therefore, must always be initialized; pointers do not have such restrictions. A pointer can be reassigned to point to different objects while a reference always refers to an object with which it was initialized.

### **What is the difference between `const char *my Pointer` and `char *const my Pointer`?**

`Const char *my Pointer` is a non-constant pointer to constant data; while `char *const my Pointer` is a constant pointer to non-constant.