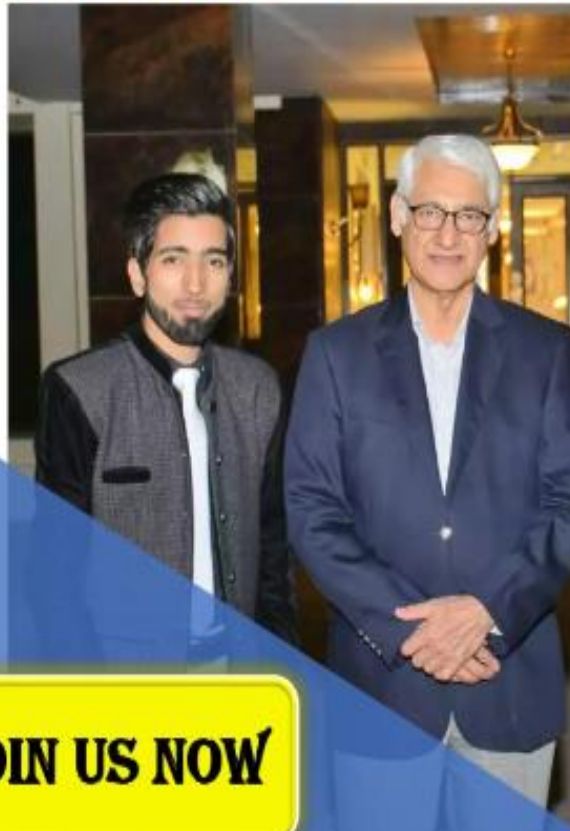


**Cs-201 Important Mcq's**  
**For Final Term !!**  
**SolveBy Vu-Topper RM!!**

وَتَعَزُّ مِنْ تَشَاءٍ وَتُذَلُّ مِنْ تَشَاءٍ



PROFESSIONAL ONLINE ACADEMY



**NOTHING IS  
IMPOSSIBLE**

**All Paid Services**

- ❖ LMS Handling
- ❖ Important Notes
- ❖ Online Classes
- ❖ Projects
- ❖ Assignments
- ❖ Quiz
- ❖ GDB's

**JOIN US NOW**

For More Info  
Contact us at:

**Rizwan Manzoor**

**0322-4021365**

**Question No:1 (Marks:1) Vu-Topper RM**

Memory allocated from heap or free store \_\_\_\_\_.

- can be returned back to the system automatically
- can be allocated to classes only
- cannot be returned back unless freed explicitly using malloc and realloc
- **cannot be returned back unless freed explicitly using free and delete operators**

**Question No:2 (Marks:1) Vu-Topper RM**

Once the \_\_\_\_\_ are created, they exist for the life time of the program

Select correct option:

- local variables
- non static variables
- **static variables**
- automatic variables

**Question No:3 (Marks:1) Vu-Topper RM**

The members of a class declared with the keyword struct are \_\_\_\_\_ by default.

Select correct option:

- static
- private
- protected
- **public.**

**Question No:4 (Marks:1) Vu-Topper RM**

If the memory in the free store is not sufficient \_\_\_\_\_.

Select correct option:

- malloc function returns 1
- malloc function returns 0
- **malloc functions returns NULL pointer**
- malloc function returns free space

**Question No:5 (Marks:1) Vu-Topper RM**

This reference to a variable can be obtained by preceding the identifier of a variable with \_\_\_\_\_.

Select correct option:

- dot operator
- **ampersand sign &**
- ^ sign
- operator

**Question No:6 (Marks:1) Vu-Topper RM**

Once an object is declared as a friend, \_\_\_\_\_.

Select correct option:

- **it has access to all non-public members as if they were public**
- it has access to public members only
- it has no access to data members of the class
- it has to protected data members only

**Question No:7 (Marks:1) Vu-Topper RM**

Reference variables must \_\_\_\_\_.

Select correct option:

- not be initialized after they are declared
- **be initialized after they are declared**
- contain integer value
- contain zero value

**Question No:8 (Marks:1) Vu-Topper RM**

The syntax of declaration of a function that returns the reference to an integer is \_\_\_\_\_.

Select correct option:

- **int &myfunc() ;**
- int myfunc();
- int myfunc() &;
- integer &myfunc();

**Question No:9 (Marks:1) Vu-Topper RM**

Reference is not really an address it is \_\_\_\_\_.

Select correct option:

- **a synonym**
- an antonym
- a value
- a number

**Question No:10 (Marks:1) Vu-Topper RM**

If the request of new operator is not fulfilled due to insufficient memory in the heap \_\_\_\_\_.

Select correct option:

- the new operator returns 2
- the new operator returns 1
- **the operator returns 0**
- free operator returns nothing

**Question No:11 (Marks:1) Vu-Topper RM**

Functions declared with the \_\_\_\_\_ specifier in a class member list are called friend functions of that class.

Select correct option:

- protected
- private
- Public
- **friend**

**Question No:12 (Marks:1) Vu-Topper RM**

The friend keyword provides access \_\_\_\_\_.

Select correct option:

- **in one direction only**
- in two directions
- to all classes
- to the data members of the friend class only

**Question No:13 (Marks:1) Vu-Topper RM**

References cannot be uninitialized. Because it is impossible to \_\_\_\_\_  
Select correct option:

- reinitialize a pointer
- **reinitialize a reference**
- initialize a NULL pointer
- cast a pointer

**Question No:14 (Marks:1) Vu-Topper RM**

15- new operator can be used for \_\_\_\_\_.

Select correct option:

- only integer data type
- only char and integer data types
- **integer , float, char and double data types**
- dot operator

**Question No:15 (Marks:1) Vu-Topper RM**

The destructor is used to \_\_\_\_\_.

Select correct option:

- allocate memory
- **deallocate memory**
- create objects
- allocate static memory

**Question No:16 (Marks:1) Vu-Topper RM**

If we want to allocate memory to an array of 5 integers dynamically, the syntax will be \_\_\_\_\_.

Select correct option:

- **int \*iptr ;iptr = new int[5] ;**
- integer iptr\*\* ;iptr= new int[5]
- int iptr ;iptr= int [5]
- iptr= new[5]

**Question No:17 (Marks:1) Vu-Topper RM**

Memory allocated from heap or free store \_\_\_\_\_.

Select correct option:

- can be returned back to the system automatically
- can be allocated to classes only
- cannot be returned back unless freed explicitly using malloc and realloc
- **cannot be returned back unless freed explicitly using free and delete operators**

**Question No:18 (Marks:1) Vu-Topper RM**

Operator overloading is to allow the same operator to be bound to more than one implementation, depending on the types of the \_\_\_\_\_.

Select correct option:

- Compilers
- **Operands**
- Function names
- Applications

**Question No:19 (Marks:1) Vu-Topper RM**

The operator to free the allocated memory using new operator is \_\_\_\_\_.

Select correct option:

- free
- del
- **delete**
- remove

**Question No:20 (Marks:1) Vu-Topper RM**

The concept of friend function negates the concept of \_\_\_\_\_.

Select correct option:

- inheritance
- polymorphism
- persistence
- **encapsulation**

**Question No:21 (Marks:1) Vu-Topper RM**

Functions declared with the \_\_\_\_\_ specifier in a class member list are called friend functions of that class.

- protected
- private
- public
- friend**

**Question No:22 (Marks:1) Vu-Topper RM**

Public or private keywords can be \_\_\_\_\_

- written only for once in the class or structure declaration
- written multiple times in the class or structure declaration**
- written only twice in the class declaration
- written outside the class

**Question No:23 (Marks:1) Vu-Topper RM**

The friend keyword provides access \_\_\_\_\_.

- in one direction only**
- in two directions
- to all classes
- to the data members of the friend class only

**Question No:24 (Marks:1) Vu-Topper RM**

References cannot be uninitialized. Because it is impossible to \_\_\_\_\_

- reinitialize a pointer
- reinitialize a reference**
- initialize a NULL pointer
- cast a pointer

**Question No:25 (Marks:1) Vu-Topper RM**

new operator can be used for \_\_\_\_\_.

- only integer data type
- only char and integer data types
- integer , float, char and double data types**
- dot operator

**Question No:26 (Marks:1) Vu-Topper RM**

The destructor is used to \_\_\_\_\_.

- allocate memory
- deallocate memory**
- create objects
- allocate static memory

**Question No:27 (Marks:1) Vu-Topper RM**

Reference is not really an address it is \_\_\_\_\_.

➤ **a synonym**

- an antonym
- a value
- a number

**Question No:28 (Marks:1) Vu-Topper RM**

If we want to allocate memory to an array of 5 integers dynamically, the syntax will be \_\_\_\_\_.

➤ **int \*iptr ;iptr = new int[5] ;**

- integer iptr\*\* ;iptr= new int[5]
- int iptr ;iptr= int [5]
- iptr= new[5]

**Question No:29 (Marks:1) Vu-Topper RM**

Memory allocated from heap or free store \_\_\_\_\_.

- can be returned back to the system automatically
- can be allocated to classes only
- cannot be returned back unless freed explicitly using malloc and realloc
- **cannot be returned back unless freed explicitly using free and delete operators**

**Question No:30 (Marks:1) Vu-Topper RM**

Operator overloading is to allow the same operator to be bound to more than one implementation, depending on the types of the \_\_\_\_\_.

- Compilers
- **Operands**
- Function names
- Applications

**Question No:31 (Marks:1) Vu-Topper RM**

The operator to free the allocated memory using new operator is \_\_\_\_\_.

- free
- del
- **delete**
- remove

**Question No:32 (Marks:1) Vu-Topper RM**

reference is a thing by which we can create \_\_\_\_\_ of any data type.

Select correct option:

- copy
- **Synonym**
- Pointer
- None of the given

**Question No:33 (Marks:1) Vu-Topper RM**

What is the sequence of event(s) when deallocating memory of an object using delete operator?

Select correct option:

- Only block of memory is deallocated for object
- Only destructor is called for object
- **Memory is deallocated first before calling destructor**
- Destructor is called first before deallocating memory

**Question No:34 (Marks:1) Vu-Topper RM**

When operator function is implemented as member function then return type of function \_\_\_\_\_.

Select correct option:

- Must be an object of same class
- Must be user-defined data type
- **Must be built-in data type**
- Can be any data type

**Question No:35 (Marks:1) Vu-Topper RM**

Overloaded new operator function takes parameter of type size\_t and returns

Select correct option:

- void (nothing)
- void pointer
- **object pointer**
- int pointer

**Question No:36 (Marks:1) Vu-Topper RM**

This reference to a variable can be obtained by preceding the identifier of a variable with \_\_\_\_\_.

Select correct option:

- dot operator
- **ampersand sign &**
- ^ sign
- \* operator

**Question No:37 (Marks:1) Vu-Topper RM**

In overloading the assignment (=) operator, which object(s) will call the operator function?  
Select correct option:

- Left object of the assignment operator
- **Right object of the assignment operator**
- Both objects will call the operator function
- No object will call the operator function

**Question No:38 (Marks:1) Vu-Topper RM**

We can \_\_\_\_\_ pointer.

Select correct option:

- increment
- decrement
- reassign
- **all of the given**

**Question No:39 (Marks:1) Vu-Topper RM**

In overloading the assignment (=) operator, which object will be passed as an argument(s) in the operator function?

Select correct option:

- **Left object of the assignment operator**
- Right object of the assignment operator
- Both objects will be passed
- No objects will be passed

**Question No:40 (Marks:1) Vu-Topper RM**

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

Select correct option:

- **True**
- False

**Question No:41 (Marks:1) Vu-Topper RM**

If class A is a friend of class B, and class B is a friend of class C. If class A wants class C to be a friend, \_\_\_\_\_

Select correct option:

- **it has to declare, class C as a friend**
- it has to declare, class B as a friend
- it has to declare, class A as a friend
- it has to declare, class B and class A as friend classes

**Question No:42 (Marks:1) Vu-Topper RM**

The destructor is used to \_\_\_\_\_.

- **deallocate memory**

**Question No:43 (Marks:1) Vu-Topper RM**

The main advantage of macros is that \_\_\_\_\_ and program runs faster

- **macros are more efficient**

**Question No:44 (Marks:1) Vu-Topper RM**

\_\_\_\_\_ is a special type of pointer we have to cast it before we use it.

➤void

**Question No:45 (Marks:1) Vu-Topper RM**

Programmer should be very careful about the memory management because it can

\_\_\_\_\_.

➤cause problems of memory leakage and dangling pointers

**Question No:46 (Marks:1) Vu-Topper RM**

Constructor is special type of function :

➤which has no return type

**Question No:47 (Marks:1) Vu-Topper RM**

Unix operating system was written in C language

➤C

**Question No:48 (Marks:1) Vu-Topper RM**

When accessing a structure member, the identifier to the left of the dot operator is the name of

\_\_\_\_\_.

➤Structure variable

**Question No:49 (Marks:1) Vu-Topper RM**

Constructor has \_\_\_\_\_.

➤the same name as of class

**Question No:50 (Marks:1) Vu-Topper RM**

For console input and output we use \_\_\_\_\_.

➤conioheaderfile

**Question No:51 (Marks:1) Vu-Topper RM**

We should not use such variable names that are starting with \_\_\_\_\_ because in C++, there are lots of internal constants and symbolic names that start with it.

➤upper case alphabets

**Question No:52 (Marks:1) Vu-Topper RM**

The members of a class declared with the keyword struct are \_\_\_\_\_ by default.

➤public

**Question No:53 (Marks:1) Vu-Topper RM**

\_\_\_\_\_ will return the number of bytes reserved for a variable or data type.

- sizeof operator
- free operator
- void pointer
- new operator

**Question No:54 (Marks:1) Vu-Topper RM**

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

- true
- false

**Question No:55 (Marks:1) Vu-Topper RM**

While using \_\_\_\_\_ operator we do not need to supply number of bytes allocated.

- insertion
- dot
- malloc
- new

**Question No:56 (Marks:1) Vu-Topper RM**

When an operator function is defined as member function for a binary Plus (+) operator then the number of argument it take is/are.

- 1
- 2
- 0
- N

**Question No:57 (Marks:1) Vu-Topper RM**

For binary member operators, operands on the \_\_\_\_\_ drives (calls) the operation.

- left
- right
- both

**Question No:58 (Marks:1) Vu-Topper RM**

Reference is not really an address it is \_\_\_\_\_

- an antonym
- a value
- a number
- synonym

**Question No:59 (Marks:1) Vu-Topper RM**

We cannot increment \_\_\_\_\_.

- pointers
- arrays
- references
- variable

**Question No:60 (Marks:1) Vu-Topper RM**

how many arguments are used while using unary operator function.

- 1
- 2
- 0
- 3

**Question No:61 (Marks:1) Vu-Topper RM**

Pointers are a special type of \_\_\_\_\_ in which a memory address is stored

Select correct option:

- variables
- Location
- Characters
- None of the given

**Question No:62 (Marks:1) Vu-Topper RM**

If x is declare as an integer, what will be result of the given expression?  $x = 2 + 3 * 5 \% ( 2 + 2 ) / 2$

Select correct option:

- 4
- 3.5
- 3
- 3.75

**Question No:63 (Marks:1) Vu-Topper RM**

Which of the following is not a valid variable name in C++?

Select correct option:

- NameOfStudent
- nameOfStudent
- \_studentName
- @studentName

**Question No:64 (Marks:1) Vu-Topper RM**

What will be the result of arithmetic expression  $6+48/4*3$ ?

Select correct option:

- 10
- 40.5
- 42
- 41

**Question No:65 (Marks:1) Vu-Topper RM**

< , <= , > , >= are called \_\_\_\_\_ operators.

Select correct option:

- Arithmetic
- Logical
- Relational
- Conational

**Question No:66 (Marks:1) Vu-Topper RM**

Pointers work by pointing to a particular \_\_\_\_\_

Select correct option:

- Value
- variable
- data type
- None of the given

**Question No:67 (Marks:1) Vu-Topper RM**

The most suitable data type for number 325.25 is \_\_\_\_\_.

Select correct option:

- char
- int
- short
- **float**

**Question No:68 (Marks:1) Vu-Topper RM**

How many times the following loop will execute? `int j = 3; while(j > 0) { cout << "Statements" << endl; j -= 2; }`

Select correct option:

- 0
- **1**
- 2
- 3

**Question No:69 (Marks:1) Vu-Topper RM**

Loops are \_\_\_\_\_ Structure

Select correct option:

- Decision
- Sequential
- **Repetition**
- None of the given options

**Question No:70 (Marks:1) Vu-Topper RM**

Structures use \_\_\_\_\_ allocation.

Select correct option:

- Queue
- **Heap**
- Cache
- Stack

**Question No:71 (Marks:1) Vu-Topper RM**

The members of a class declared with the keyword struct are \_\_\_\_\_ by default.

- **public**
- private

**Question No:72 (Marks:1) Vu-Topper RM**

\_\_\_\_\_ will return the number of bytes reserved for a variable or data type.

- sizeof operator
- free operator
- void pointer
- **new operator**

**Question No:73 (Marks:1) Vu-Topper RM**

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

- true
- false

**Question No:74 (Marks:1) Vu-Topper RM**

While using \_\_\_\_\_ operator we do not need to supply number of bytes allocated.

- insertion
- dot
- malloc
- new

**Question No:75 (Marks:1) Vu-Topper RM**

5. When an operator function is defined as member function for a binary Plus (+) operator then the number of argument it take is/are.

- 1
- 2
- 0
- N

**Question No:76 (Marks:1) Vu-Topper RM**

For binary member operators, operands on the \_\_\_\_\_ drives (calls) the operation.

- left
- right
- both

**Question No:77 (Marks:1) Vu-Topper RM**

We can \_\_\_\_\_ references.

- reassign
- join

**Question No:78 (Marks:1) Vu-Topper RM**

Reference is not really an address it is \_\_\_\_\_

- an antonym
- a value
- a number
- synonym

**Question No:79 (Marks:1) Vu-Topper RM**

We cannot increment \_\_\_\_\_.

- pointers
- arrays
- references
- variable

**Question No:80 (Marks:1) Vu-Topper RM**

how many arguments are used while using unary operator function.

- 1
- 2
- **0**
- 3

**Question No:81 (Marks:1) Vu-Topper RM**

int &i; It means that i is a \_\_\_\_\_ to an integer.

- pointer
- **reference**
- data type
- none

**Question No:82 (Marks:1) Vu-Topper RM**

We can \_\_\_\_\_ references.

- increment
- decrement
- reassign
- **non**

**Question No:83 (Marks:1) Vu-Topper RM**

\_\_\_\_\_ will return the number of bytes reserved for a variable or data type.

- sizeof operator
- free operator
- **new operator**

**Question No:84 (Marks:1) Vu-Topper RM**

For binary member operators, operands on the \_\_\_\_\_ drives (calls) the operation.

- left
- **right**
- both
- none

**Question No:85 (Marks:1) Vu-Topper RM**

int &i; It means that i is a \_\_\_\_\_ to an integer.

- pointer
- **reference**

**Question No:86 (Marks:1) Vu-Topper RM**

An address is a \_\_\_\_\_, while a pointer is a \_\_\_\_\_.

- **constant , variable**
- variable , constant
- global , variable
- non static variable , constant

**Question No:87 (Marks:1)****Vu-Topper RM**

A pointer is \_\_\_\_\_.

- **the address of a variable**
- an indication of the variable to be accessed next
- a variable for storing address
- the data type of an address variable

**Question No:88 (Marks:1)****Vu-Topper RM**

The new operator automatically determines the size of memory required to store that object , so it does not need \_\_\_\_\_.

- free operator
- dot operator
- **sizeof operator**
- size off operator

**Question No:89 (Marks:1)****Vu-Topper RM**

We cannot increment \_\_\_\_\_.

- pointers
- arrays
- **refrence**

**Question No:90 (Marks:1)****Vu-Topper RM**

The \_\_\_\_\_ data type always represents an empty set of values in C++

- nil
- **void**
- zero

**Question No:91 (Marks:1)****Vu-Topper RM**

public or private keywords can be \_\_\_\_\_

- **written only for once in the class or structure declaration**
- written multiple times in the class or structure declaration
- written only twice in the class declaration
- written outside the class

**Question No:92 (Marks:1)****Vu-Topper RM**

The destructor is used to \_\_\_\_\_.

- allocate memory
- **deallocate memory**
- create objects

**Question No:93 (Marks:1)****Vu-Topper RM**

Which one of the following is the declaration of overloaded pre-increment operator implemented as member function?

- Class-name operator +() ;
- Class-name operator +(int) ;
- Class-name operator ++();
- **Class-name operator ++(int) ;**

**Question No:94 (Marks:1) Vu-Topper RM**

You cannot overload the \_\_\_\_\_ operator

- ?:
- \*
- /
- +

**Question No:95 (Marks:1) Vu-Topper RM**

In C language, the region of memory allocated at runtime is called \_\_\_\_\_.

- stack
- heap**
- free store
- available memory

**Question No:96 (Marks:1) Vu-Topper RM**

Reference variables must \_\_\_\_\_.

- not be initialized after they are declared
- be initialized after they are declared**
- contain integer value
- contain zero value

**Question No:97 (Marks:1) Vu-Topper RM**

reference is a thing by which we can create \_\_\_\_\_ of any data type.

- copy
- Synonym**
- Pointer
- none

**Question No:98 (Marks:1) Vu-Topper RM**

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

- true**
- false

**Question No:99 (Marks:1) Vu-Topper RM**

In functions, that returns reference use \_\_\_\_\_

- global or local variables**
- global or static variables
- ordinary variables
- array

**Question No:100 (Marks:1) Vu-Topper RM**

The friend functions are

- not member of a class
- member of a class**
- inline functions
- void functions

**Question No:101 (Marks:1) Vu-Topper RM**

The concept of \_\_\_\_\_ allows us to separate the interface from the implementation of the class.

- pointers
- inheritance
- polymorphism
- **encapsulation**

**Question No:102 (Marks:1) Vu-Topper RM**

\_\_\_\_\_ operators are the ones that require only one operator to work.

- Unit
- **Unary \***
- Single
- none

**Question No:103 (Marks:1) Vu-Topper RM**

C++ offers \_\_\_\_\_ levels of data access control inside a class

- **three \***
- four
- two
- five

**Question No:104 (Marks:1) Vu-Topper RM**

References cannot be uninitialized. Because it is impossible to \_\_\_\_\_

- reinitialize a pointer
- **reinitialize a reference**
- initialize a NULL pointer
- cast a pointer

**Question No:105 (Marks:1) Vu-Topper RM**

In C language, the region of memory allocated at runtime is called \_\_\_\_\_

- stack
- **heap**
- free store
- available memory

**Question No:106 (Marks:1) Vu-Topper RM**

In functions, that returns reference use \_\_\_\_\_

- global or local variables
- **global or static variables**
- ordinary variables
- array

**Question No:107 (Marks:1) Vu-Topper RM**

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

- **true**
- false

**Question No:108 (Marks:1) Vu-Topper RM**

public or private keywords can be \_\_\_\_\_

- written only for once in the class or structure declaration
- written multiple times in the class or structure declaration
- written only twice in the class declaration
- **written outside the class**

**Question No:109 (Marks:1) Vu-Topper RM**

The friend function of a class can have access \_\_\_\_\_

- **to the public data members only**
- to the private data members
- to the protected data members
- to the main program

**Question No:110 (Marks:1) Vu-Topper RM**

In c++ the region of available memory is called \_\_\_\_\_

- **heap**
- empty space
- free store
- allocated memory

**Question No:111 (Marks:1) Vu-Topper RM**

When an operator function is defined as member function for a binary Plus (+) operator then the number of argument it take is/are

- zero
- **one**
- twon
- n arguments

**Question No:112 (Marks:1) Vu-Topper RM**

If class A is a friend of class B, and class B is a friend of class C. If class A wants class C to be a friend, \_\_\_\_\_

- it has to declare, class C as a friend
- **it has to declare, class B as a friend**
- it has to declare , class A as a friend
- it has to declare, class B and class A as friend classes

**Question No:113 (Marks:1) Vu-Topper RM**

We can \_\_\_\_\_reference

- increment
- decrement
- reassign
- **none of the given**

**Question No:114 (Marks:1) Vu-Topper RM**

A pointer is

- **the address of a variable**

**Question No:115 (Marks:1) Vu-Topper RM**  
An indication of the variable to be accessed next. a variable for storing address  
➤ **the data type of an address variable**

**Question No:116 (Marks:1) Vu-Topper RM**  
Overloaded member operator function is always called by \_\_\_\_\_  
➤ Class  
➤ **Object**  
➤ Compiler  
➤ Primitive data type

**Question No:117 (Marks:1) Vu-Topper RM**  
The difference between pointers and references is that  
➤ we cannot do arithmetic with pointers  
➤ **we can do arithmetic with pointers**  
➤ we cannot reassign pointers  
➤ we can assign references

**Question No:118 (Marks:1) Vu-Topper RM**  
While using \_\_\_\_\_ operator we do not need to supply number of bytes allocated.  
➤ insertion  
➤ dot  
➤ malloc  
➤ **new**

**Question No:119 (Marks:1) Vu-Topper RM**  
Memory allocated from heap or free store  
➤ can be returned back to the system automatically  
➤ can be allocated to classes only  
➤ cannot be returned back unless freed explicitly using malloc and realloc  
➤ **cannot be returned back unless freed explicitly using free and delete operators**

**Question No:120 (Marks:1) Vu-Topper RM**  
If B is designated as friend of A, B can access A's non-public members  
➤ B cannot access private member of A  
➤ **B cannot access protected member of A**  
➤ A can access non-public members of B  
➤ A cannot access B

**Question No:121 (Marks:1) Vu-Topper RM**  
Which one of the following is the declaration of overloaded pre-increment operator implemented as member function?  
➤ **Class-name operator +() ;**  
➤ Class-name operator +(int) ;  
➤ Class-name operator ++();  
➤ Class-name operator ++(int) ;

**Question No:122 (Marks:1) Vu-Topper RM**

An address is a \_\_\_\_\_, while a pointer is a

- **constant , variable**
- variable , constant

**Question No:123 (Marks:1) Vu-Topper RM**

To prevent dangling reference the functions returning reference should be used with

- **local variables**
- global variables only.
- arrays

**Question No:124 (Marks:1) Vu-Topper RM**

When an operator function is defined as member function for a binary Plus (+) operator then the number of argument it take is/are.

Select correct option:

- Zero
- **One**
- Two
- N arguments

**Question No:125 (Marks:1) Vu-Topper RM**

Deleting an array of objects without specifying [] brackets may lead to memory leak.

Select correct option:

- True
- **False**

**Question No:126 (Marks:1) Vu-Topper RM** Which of the following syntax is best used to delete an array of 5 objects named 'string' allocated using new operator.

Select correct option:

- delete string ;
- **delete []string ;**
- delete string[] ;
- delete string[5] ;

**Question No:127 (Marks:1) Vu-Topper RM**

The function will return a reference to the global variable that exists throughout the program and thus there will be no danger of \_\_\_\_\_.

Select correct option:

- garbage collection
- **dangling reference**
- wastage of memory
- system crash

**Question No:128 (Marks:1) Vu-Topper RM**

What is the sequence of event(s) when deallocating memory of an object using delete operator?

Select correct option:

- Only block of memory is deallocated for object
- Only destructor is called for object
- **Memory is deallocated first before calling destructor**
- Destructor is called first before deallocating memory

**Question No:129 (Marks:1) Vu-Topper RM**

Overloaded new operator function takes parameter of type size\_t and returns

Select correct option:

- void (nothing)
- void pointer
- **object pointer**
- int pointer

**Question No:130 (Marks:1) Vu-Topper RM**

We can delete an array of objects without specifying [] brackets if a class is not doing dynamic memory allocation internally.

Select correct option:

- **True**
- False

**Question No:131 (Marks:1) Vu-Topper RM**

What is the function of the following statement to delete an array of 5 objects named 'arr' allocated using new operator? delete arr ;

Select correct option:

- Deletes all the objects of array
- Deletes only one object of array
- Do not delete any object
- **Results into syntax error**

**Question No:132 (Marks:1) Vu-Topper RM**

Friend function declaration can go \_\_\_\_\_ the class

Select correct option:

- only outside
- **only within**
- in the member function of
- anywhere in