

CS201 FOR FINAL TERM

UPDATED QUIZ 2 2021

PROVIDE BY ORANGE MONKEY

1. To avoid dangling reference don't return

The reference of a local variable from the function.....confirm

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

Class name operator ++()....confirm from net

3. Reference is a thing by which we can create _____ of any data type

Synonym.....confirm

4. Reference cannot be uninitialized because it is impossible to _____.

Initialize a Null pointer.....confirm

5. The syntax of declaration of a function that returns the reference to an integer is _____.

Int & myfunc().....confirm

6. Friend function declaration can go _____ the class

Anywhere in.....confirm

7. To prevent dangling reference the functions returning reference should be used with _____.

Static and global variables.....confirm

8. The only operator that the compiler overloads for user define data type by default in

Arrangement (=) operator confirm from net

9. If operator function is non-member function then object on left side of operator cannot be _____?

Object of member function.....confirm

10. Which of the following functionality can be achieved through overloading?

New operators cannot be defined through operator overloading

11. The return type of the overloading new operator must be?

Void*....confirm from net

12. For binary member operators, operands on the _____ derives (calls) the operation.

Left.....confirm

13. Identify the correct method of adding two strings, s1 and s2.

Strcat(s1,s2)

14. A friend function of a class is a function defined _____.

Outside that class and that has the right to access all members of the class

15. Care must be taken about the correct _____ of operator while overloading.

Both Semantic and Complexity.....confirm from net

16. We can _____ reference.

None of the given....confirm

17. If text is a pointer of type string then what will be the functionality of following statement?

Text = new String {5};

Creates array of 5 objects dynamically.....confirm from net

18. Look at the statement given below

Int & a; and tell what will happen?

Compiler will generate an error 'a' declared as reference but not initialized.....confirm

19. The syntax of declaration of a function that returns the reference to an integer is _____.

Int & myfunc();....confirm

20. Which of the following syntax is best used to delete an array of 5 objects named 'string' allocated using new operator.

Delete []string;....confirm from net

21. If we do not write our own assignment operator then which of the following problem may occur?

22. Identify the correct syntax for making a class friend of other class

```
friend ClassOne
{
    OtherClass;
    private;
    //here we write the data members of ClassOne
};
```

23. Once an object is declared as a friend _____

It has access to all non-public members as if they were public....confirm from net

24. Reference is a thing by which we can create _____ of any data type.

alias or synonym...both.....confirm

25. A _____ function of a class is defined outside that class scope. But it can access all private and protected members of the class.

Firend.....confirm

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

Static variables.....confirm

27. A class can be declared as a _____ of other class.

Friend....confirm

28. Reference variables must _____.

Be initialized after they are declared....confirm from net

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

True

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

Operands.....confirm

31. The friend keyword provides access _____.

In one direction only.....confirm

32. The friend functions are _____.

Not member of a class....confirm

33. Ternary operator is shown as _____.

?;.....confirm

34. Reference is not really an address it is _____.

A synonym

35. In C++ operators, which of the following operator cannot be overloaded _____.

?;.....confirm

36. Overloaded assignment operator must be

Member function of class

37. An address is a _____, while a pointer is a _____.

Variable, constant...confirm from net

38. When new operator is overloaded at global level then corresponding built-in new operator will also be visible to whole of the program.

False....confirm

39. When new or delete operator is overloaded at global level then corresponding built-in new or delete operator will not be visible to whole of the program.

True....confirm

40. When an operator function is defined as member function for a binary Plus (+) operator then the number of extra arguments it takes is/are

One....confirmF

41. $i+=2$ is equivalent to _____.

$i = i + 2;$

42. Assignment operator is used to initialize a newly declared object from existing object.

True....confirm from net

43. When an array of object is created dynamically then there is no way to provide parameterized constructors for array of objects.

True...confirm from net

44. In overloading the assignment (=) operator, which object will be passed as an arguments in the operator function?

Left object of the assignment operator

45. Friend functions are _____

Private

46. We can _____ pointer.

Decrement increment reassign...all options..... confirm

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

Dangling reference.....confirm

48. The reference data types are used as _____ variables without any _____ operator.

Ordinary , deference.....confirm

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

Delete arr;

Do not delete any object

50. If B is designated as friend of A, B can access A's non-public members

A cannot access B....confirm from net

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

Memory is deallocated first before calling destructor...confirm from net

52. _____ operators are the ones that require two operands on both sides of the operator.

Binary.....confirm

53. Int & I; it means that i is a _____ to an integer.

Reference.....confirm

54. When operator function is implemented as member function then return type of function _____.

Can be any data type....confirm from net

55. The concept of _____ allows us to separate the interface from the implementation of the class.

Encapsulation....confirm

56. With user data type variables (objects) self assignment can produce

Logical error....confirm

57. When the compiler overloads the assignment (=) operator by default then

Compiler does member wise assignment..confirm from net

58. Which of the following is unary operator?

+i, i++, ==I ...all options.....confirm

59. Reference is not really an address it is _____.

A synonym

60. What is meant by the following statement?

String str[5] = {String("Programming"), String("CS201")};

Parameterized constructor will be called for all objects of array

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

Two...confirm

62. Friend classes are used in cases where one class is _____ to another class

Independent

63. The difference between pointers and references is that _____

We can do arithmetic with pointers....confirm

64. When an operator function is defined as member function for a Unary operator then the number of extra arguments it takes is/are.

One.....confirm

65. *This is a pointer that always points to

Current pointer of the class

Question # 1:-

Care must be taken about the correct _____ of operator while overloading. (Choose the most appropriate).

1. Semantic
2. Complexity
3. Both Semantic and Complexity
4. None of the given options

Question # 2:-

In functions, that returns reference use _____.

1. global or local variables
2. global or static variables
3. ordinary variables

Question # 3:-

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

1. Left object of the assignment operator
2. Right object of the assignment operator
3. Both objects will call the operator function
4. No object will call the operator function

Question # 4:-

Friend functions are _____.

1. Unidirectional (not confirm)
2. bidirectional
3. like inline functions
4. private

Question # 5:-

We cannot increment _____.

1. pointers
2. arrays
3. references
4. Variables

Question # 6:-

A pointer is _____.

1. the address of a variable
2. an indication of the variable to be accessed next.
3. the data type of an address variable

Question # 7:-

Overloaded assignment operator must be

1. Member function of class
2. Non-member function of class (not confirm)
3. Friend function of class
4. Global function

Question # 8:-

The concept of friend function negates the concept of _____.

1. inheritance
2. polymorphism
3. persistence
4. encapsulation

Question # 9:-

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, _____

1. it has to declare, class C as a friend
2. it has to declare, class B as a friend
3. it has to declare , class A as a friend
4. it has to declare, class B and class A as friend classes

Question # 10:-

The difference between pointers and references is that _____.

1. we cannot do arithmetic with pointers

2. we can do arithmetic with pointers
3. we cannot reassign pointers
4. we can assign references

Question # 11:-

A reference cannot be _____.

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

Question # 12:-

An address is a _____, while a pointer is a _____.

1. constant , variable
2. variable , constant
3. global , variable
4. non static variable , constant

Question # 13:-

The reference data types are used as _____ variables without any _____ operator

1. ordinary , deference
2. global , dot
3. static , deference
4. local , &

Question # 14:-

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();`

Question # 15:-

References cannot be uninitialized. Because it is impossible to

1. reinitialize a pointer
 2. reinitialize a reference
 3. **initialize a NULL pointer**
- cast a pointer

If we write a statement like `s2 = s1;` ____ will be the calling object and ____ will be passed to the = operator as an argument.

- ▶ `s1, s1`
- ▶ `s1, s2`
- ▶ **`s2, s1`**
- ▶ `s2, s2`

If we write a statement like `s2 = s1;` `s2` will be the calling object and `s1` will be passed to the = operator as an argument. P# 397

www.vuzs.info

what will be out put of `cout << setfill('0') << setw(7) << 128;`

Overloaded new operator function takes parameter of type `size_t` and returns

- ▶ **void (nothing)**

- ▶ **void pointer**
- ▶ object pointer
- ▶ int pointer

Also note that the *new* operator returns a *void* pointer. Any *new* operator we write must have this parameter and return type.

Which of the following is the correct way to declare a variable x of integer type?

- ▶ x int ;
- ▶ integer x ;
- ▶ **int x;**
- ▶ x integer

Reserve words cannot be used as a variable name.

- ▶ **True**
- ▶ False

There are few data types in C language. These data types are reserved words of C language. The reserve words can not be used as a variable manes. P# 17

A template function must have at least ----- generic data type

- ▶ Zero
- ▶ **One**
- ▶ Two
- ▶ Three

The function arguments must contain at least one generic data type. P# 499

Template functions can also be overloaded

- ▶ True
- ▶ **False**

We can write overloaded template functions as long as there is use of different number or type of arguments.. P # 503

We can not make a member function of a class as template function.

- ▶ True
- ▶ **False** not sure

When break statement is encountered in switch statement, it

- ▶ Stops the entire program
- ▶ **Stops the execution of current statement**
- ▶ Exits from switch statement
- ▶ None of the given options

We can also define a variable of user define data type (object) as static.

- ▶ True
- ▶ False

The declarator of Plus (+) member operator function is

- ▶ **Class-Name operator + (Class-Name rhs)**
- ▶ operator Class-Name + ()
- ▶ operator Class-Name + (rhs)
- ▶ Class-Name operator + ()

Let suppose

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

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

This can be interpreted by the compiler as:

- ▶ a = (b = (c = (d = (e = 42))));
 - ▶ (a = b = (c = (d = (e = 42))));
 - ▶ a = b = (c = (d = (e = 42)));
 - ▶ (a = b) = (c = d) = (e = 42);
- ```
a = (b = (c = (d = (e = 42)))) ;
```

What will be the range of numbers generated by function rand () % 9?

- ▶ 0 to 9
- ▶ 1 to 9
- ▶ 0 to 8
- ▶ 1 to 8

When 6 divides any number, the remainder will always be less than 6. Ther result will be between therefore we will add 1. 1 + rand ( ) % 6;

Which of the following is the correct function call having array named *student* of 10 elements as a parameter.

- ▶ **addRecord(student[]) ;**
- ▶ addRecord(student) ;
- ▶ addRecord(student[10]) ;
- ▶ addRecord(\*student) ;

when we pass array we don't give limit of array

Example:

**Pass array to function**

---

```

#include
#include
void read(int *,int);
void dis(int *,int);

void main()
{
 int a[5],b[5],c[5],i;

 printf("Enter the elements of first list \n");
 read(a,5);
 printf("The elements of first list are \n");
 dis(a,5);
}

void read(int c[],int i)
{
 int j;
 for(j=0;j<i;j++) scanf("%d",&c[j]);
 fflush(stdin);
}

void dis(int d[],int i)
{
 int j;
 for(j=0;j<i;j++) printf("%d ",d[j]);
 printf("\n");
}

```

Declaring structures does not mean that memory is allocated.

Example:

- ▶ True
- ▶ False

**structures do not occupy any memory until it is associated with the structure variable**

Identifier is a name that can be given to variables, labels and functions.

- ▶ True
- ▶ False

**An 'Identifier' means any name that the user creates in his/her program. These names can be of variables, functions and labels**

If a class A declares itself a friend of class B and a class B declares itself a friend of class C then

- ▶ Class A is also a friend of class C.
- ▶ Class B is also a friend of class A.
- ▶ **Class A is also a friend of class C if A declares C as its friend.**
- ▶ Class A is also a friend of class C if C declares A as its friend.

If we want a two-way relationship, OtherClass will have to declare ClassOne as a friend class, resulting in a complete two-way relationship

Which of the following statement is best regarding declaration of friend function?

- ▶ Friend function must be declared after public keyword.
- ▶ Friend function must be declared after private keyword.
- ▶ Friend function must be declared at the top within class definition.
- ▶ It can be declared anywhere in class as these are not affected by the public and private keywords.

*Friend is a very strong statement. It is too strong to be affected by public or private we can put it anywhere in the class*

A pointer is a special type of variable that contain \_\_\_\_\_

- ▶ Memory Address
- ▶ Data values
- ▶ Both Values and Memory
- ▶ None of given of options

*Pointer is a special type of variable that contains a memory address.*

When memory for a program is allocated at run time then it is called \_\_\_\_\_

- ▶ static memory allocation
- ▶ dynamic memory allocation
- ▶ stack memory allocation
- ▶ virtual memory allocation

*When we create an object of the class at run time, it will allocate memory according to our requirement. So there is no waste of memory and the situations in which we want to store large data in small memory or vice versa are prevented. So we do dynamic memory allocation inside these classes.*

What purpose do classes serve?

- ▶ Data encapsulation
- ▶ Providing a convenient way of modeling real-world objects
- ▶ Simplifying code reuse
- ▶ All of the given options

Which of the following function cannot be overloaded?

- ▶ Member functions
- ▶ Utility functions
- ▶ Constructor

### ► Destructor

*The destructors can be summarized as The destructors cannot be overloaded. The destructors take no arguments. The destructors don't return a value*

The following prototype of unary operator function indicates that it is \_\_\_\_\_ .

Date operator++(int )

### ► Member functions of post increment operator

- Member functions of pre increment operator
- Non-member functions of post increment operator
- Non-member functions of pre increment operator

Overloading Unary Operators

// Preincrement operator overloaded as a member function.

Date Date::operator++()

```
{
 helpIncrement();
 return *this; // value return; not a reference return
}
```

// Postincrement operator overloaded as a member function.

// Note that the dummy integer parameter does not have a  
// parameter name.

Date Date::operator++(int)

```
{
 Date temp = *this;
 helpIncrement();

 // return non-incremented, saved, temporary object
 return temp; // value return; not a reference return
} // This paper was solved by vuzs Team and meant for hosting
at vuzs otherwise its stolen contents
```

Static variable which is defined in a function is initialized \_\_\_\_\_.

### ► Only once during its life time

- Every time the function call
- Compile time of the program
- None of the above

*Once the static variables are created, they exist for the life of the program. They do not die.*

In the member initialize list, the data members are initialized,

### ► From left to right

- From right to left
- In the order in which they are defined within class
- None of the given options

If we do not indent the code properly it will \_\_\_\_\_

- ▶ Be a syntax error
- ▶ Be a logical error
- ▶ **Not be an error at all**
- ▶ None of the given options

we indent the code for better readability and understanding

Truth tables are used for analyzing \_\_\_\_\_.

- ▶ **logical expressions**
- ▶ arithmetic expressions
- ▶ both logical and arithmetic expressions
- ▶ none of the given options.

The truth tables are very important. These are still a tool available for analyzing logical expressions.

Static memory allocation is also known as \_\_\_\_\_

- ▶ Dynamic allocation
- ▶ **Compile time allocation**
- ▶ Run time allocation
- ▶ None of the given options

This type of memory static allocation. It is also known as compile time allocation.

**Question No: 1** ( Marks: 1 ) - Please choose one

When we define an array of objects then,

- Destructor will call once for whole array
- **Destructor will call for each object of the array**
- Destructor will never call
- Depends on the size of array

**Question No: 2** ( Marks: 1 ) - Please choose one

We can also create an array of user define data type

- **True**
- False

**Question No: 3** ( Marks: 1 ) - Please choose one

What is the sequence of event(s) when allocating memory using new operator?

- Only block of memory is allocated for objects
- Only constructor is called for objects
- **Memory is allocated first before calling constructor**
- Constructor is called first before allocating memory

If a single object is allocated, *operator new* is called to allocate memory, and then the constructor is called to initialize the object.

- If an array of objects is allocated, *operator new[]* is called to allocate memory for the whole array, and then the constructor is called for each element of the array.
- When a single object is deleted, the destructor for the object is called first, and then *operator delete* is called to free the memory occupied by the object.
- When an array of objects is deleted, the destructor for each element of the array object is called first, and then *operator delete[]* is called to free the memory occupied by the array.

<http://www.vuzs.info/>

**Question No: 4 ( Marks: 1 ) - Please choose one**

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

- **True**
- False

Although, this is good to deallocate an array of objects without specifying array operator ([]) as there is no dynamic memory allocation occurring from inside the Date class. But this is a bad practice.

**Question No: 5 ( Marks: 1 ) - Please choose one**

The declarator of Plus (+) member operator function is

- **Class-Name operator + (Class-Name rhs)**

- Operator Class-Name + ( )
- Operator Class-Name + ( rhs)
- Class-Name operator + ( )

Page 371,373 example are here

Complex operator + (Complex & );

Complex operator + (parameter-list);

The syntax of the prototype of the overloaded operator function is:

return-type operator operator-symbol (parameter-list);

operator is the keyword here. An example of this will be as follows:

Complex operator + (Complex & );

**Question No: 6 ( Marks: 1 ) - Please choose one**

The second parameter of operator functions for << and >> are objects of the class for which we are overloading these operators

- **True** (not sure)
- False

<http://www.vuzs.info/>

**Question No: 7 ( Marks: 1 ) - Please choose one**

Which of the following is correct way to initialize a variable x of int type with value 10?

- int x ; x = 10 ;
- **int x = 10 ;**
- int x, x = 10;
- **x = 10 ;**

variable already created in question only it is asking for initialization.

**Question No: 8 ( Marks: 1 ) - Please choose one**

Default mechanism of function calling in case of array is \_\_\_\_\_ and in case of variable is \_\_\_\_\_

- Call by value, call by reference
- Call by reference, call by reference
- **Call by reference, call by value**
- Call by value, call by value

**Question No: 9 ( Marks: 1 ) - Please choose one**

What does STL stand for?

- Source template library
- **Standard template library**
- Stream template library
- Standard temporary library

[STL stands for Standard Template Library](#)

**Question No: 10 ( Marks: 1 ) - Please choose one**

Skill(s) that is/are needed by programmers \_\_\_\_\_

- Paying attention to detail
- Think about the reusability
- Think about user interface

· **All of the given options**

Programming is an important activity as people life and living depends on the programs one make. Hence while programming one should

- Paying attention to detail
- Think about the reusability.
- Think about user interface
- Understand the fact the computers are stupid
- Comment the code liberally

<http://www.vuzs.info/>

**Question No: 11** ( Marks: 1 ) - Please choose one

For which array, the size of the array should be one more than the number of elements in an array?

- int
- double
- float

· **char**

**Question No: 12** ( Marks: 1 ) - Please choose one

new and delete are \_\_\_\_\_ whereas malloc and free are \_\_\_\_\_

- Functions, operators
- Classes, operators

· **Operators, functions**

- Operators, classes

new and delete are operators in c++

C functions like **malloc()** and **free()** functions can also be used from within C++ code

**Question No: 13** ( Marks: 1 ) - Please choose one

The prototype of friend functions must be written \_\_\_\_\_ the class and its definition must be written \_\_\_\_\_

- inside, inside the class

· **inside, outside the class**

- outside, inside the class
- outside, outside the class

**Question No: 14** ( Marks: 1 ) - Please choose one

Friend function of a class are \_\_\_\_\_ of a class.

· **Non-member functions** not sure

- Friend functions
- Any function outside class
- None of the given options

<http://www.vuzs.info/>

**Question No: 15** ( Marks: 1 ) - Please choose one

If overloaded plus operator is implemented as non-member function then which of the following statement will be true for the statement given below?

obj3 = obj1 + obj2 ;

· **obj2 will be passed as an argument to + operator whereas obj2 will drive the + operator**

- obj1 will drive the + operator whereas obj2 will be passed as an argument to + operator

- Both objects (obj1, obj2) will be passed as arguments to the + operator

- Any of the objects (obj1, obj2) can drive the + operator

c3 = c1 + c2 ; In the above statement ( c3 = c1 + c2; ), c1 is the object that is calling or driving

the + operator. c2 object is being passed as an argument to the + operator. So c1 and c2 objects are added by the + operator and resultant

**Question No: 16** ( Marks: 1 ) - Please choose one

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) ;

Overloading Unary Operators

// **Preincrement** operator overloaded as a member function.

```
Date Date::operator++()
{
 helpIncrement();
 return *this; // value return; not a reference return
}
```

// **Postincrement** operator overloaded as a member function.

// Note that the dummy integer parameter does not have a  
// parameter name.

```
Date Date::operator++(int)
{
 Date temp = *this;
 helpIncrement();

 // return non-incremented, saved, temporary object
 return temp; // value return; not a reference return
}
```

**Question No: 17** ( Marks: 1 ) - Please choose one

For cin, the source is normally a \_\_\_\_\_ and destination can be \_\_\_\_\_

- File, native data type
- Disk, user-define type
- **Keyboard, variable**
- File, user-define type

For cin, the source is normally keyboard and the destination can be an ordinary variable i.e. native-data type variable. It could be some area of memory or our own data type, i.e. object for which we h

**Question No: 18** ( Marks: 1 ) - Please choose one

We can do condition compilation with pre processor directives.

- **True**
- False

All the preprocessor directives start with the sharp sign (#). We can also do conditional compilation with it.

**Question No: 19** ( Marks: 1 ) - Please choose one

The programs, in which we allocate static memory, run essentially on \_\_\_\_\_

- Heap
- System Cache
- None of the given options
- **Stack**

The programs, in which we allocate static memory, run essentially on stack

**Question No: 20** ( Marks: 1 ) - Please choose one

A template function must have at least ----- or more arguments

- Zero
- **One**
- Two
- Three

The function arguments must contain at least one generic data type. Normal function declaration is: return\_type function\_name(argument\_list)

**Question No: 21** ( Marks: 1 ) - Please choose one

The default value of a parameter can be provided inside the \_\_\_\_\_

- function prototype
- function definition
- **both function prototype or function definition**
- none of the given options

The default value of a parameter is provided inside the function prototype or function definition.

**Question No: 22** ( Marks: 1 ) - Please choose one

While calling function, the arguments are assigned to the parameters from \_\_\_\_\_

- **left to right**
- right to left

- no specific order is followed
- none of the given options

While calling function, the arguments are assigned to the parameters from left to right.

**Question No: 23 ( Marks: 1 ) - Please choose one**

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**
- Two
- N arguments

**Operators as member functions**

Aside from the operators which must be members, operators may be overloaded as member or non-member functions. The choice of whether or not to overload as a member is up to the programmer. Operators are generally overloaded as members when they:

change the left-hand operand, or

1. require direct access to the non-public parts of an object.

When an operator is defined as a member, the number of explicit parameters is reduced by one, as the calling object is implicitly supplied as an operand. Thus, binary operators take one explicit parameter and unary operators none. In the case of binary operators, the left hand operand is the calling object, and no type coercion will be done upon it.

**Question No: 24 ( Marks: 1 ) - Please choose one**

new operator allocates memory from free store and return \_\_\_\_\_

- **A pointer**
- A reference
- An integer
- A float

**Question No: 25 ( Marks: 1 ) - Please choose one**

With \_\_\_\_\_ user-defined data type variables (Objects), self assignment can produce\_\_\_\_\_

- **Syntax error** not sure
- Logical error
- Link error
- Non of the given options

**Question No: 26 ( Marks: 1 ) – Write Simple Program**

Assignment operator is used to initialize a newly declared object from existing object

- **True**
- False

**Question No: 27 ( Marks: 1 ) – Briefly define/Justify**

When an object of a class is defined inside an other class then,

- Constructor of enclosing class will be called first

· **Constructor of inner object will be called first**

- Constructor and Destructor will be called simultaneously
- None of the given options

• **A class can contain instances of other classes as its data members.** • It is a way of reusing the code when we contain objects of our already written classes into a new class.

- The inner data members of the object are constructed and then the object itself.

The order of destruction of an object is reverse to this construction order, where the outer object is destroyed first before the inner data members.

- Initializer list is used to initialize the inner objects at the construction time.

• In C++, we can have structures or classes defined inside classes. Classes defined within other classes are called nested classes.

**Question No: 28 ( Marks: 1 ) – Brief answer required**

In the member initializer list, the data members are initialized,

· **From left to right**

- From right to left
- In the order in which they are defined within class
- None of the given options

**Question No: 29**

**( Marks: 1) - Brief answer required**

"new" and "delete" keywords are \_\_\_\_\_ in C++ language

· Built-in- Function

· **Operators**

· Memory Allocation Function

· None of the given options

In C/C++ if we define an array of size eight (8) i.e. `int Arr [8];` then the last element of this array will be stored at,

► `Arr[0]`

- ▶ Arr[8]
- ▶ **Arr[7]**
- ▶ Arr[-1]

Question No: 2 ( M - 1 ) .

When an array is passed to a function then default way of passing this array is,

- ▶ By data
- ▶ **By reference**
- ▶ By value
- ▶ By data type

Question No: 3 ( M - 1 ) .

Array is a data structure which store

- ▶ Memory addresses
- ▶ Variables
- ▶ Data Type
- ▶ **Data**

Question No: 4 ( M - 1 ) .

We can also create an array of user define data type.

- ▶ **True**
- ▶ False

Question No: 5 ( M - 1 ) .

When we define an array of objects then,

- ▶ Destructor will call once for whole array
- ▶ **Destructor will call for each object of the array**
- ▶ Destructor will never call
- ▶ Depends on the size of array

Question No: 6 ( M - 1 ) .

What is the sequence of event(s) when allocating memory using new operator?

- ▶ Only block of memory is allocated for objects
- ▶ Only constructor is called for objects
- ▶ **Memory is allocated first before calling constructor**
- ▶ Constructor is called first before allocating memory

Question No: 7 ( M - 1 ) .

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

- ▶ **True**
- ▶ False

Question No: 8 ( M - 1 ) .

The second parameter of operator functions for are objects of the class for which we are overloading these operators.

- ▶ **True**
- ▶ False

Question No: 9 ( M - 1 ) .

Which of the following is correct way to initialize a variable x of int type with value 10?

- ▶ int x ; x = 10 ;
- ▶ int x = 10 ;
- ▶ int x, x = 10;
- ▶ **x = 10 ;**

Question No: 10 ( M - 1 ) .

Default mechanism of function calling in case of array is \_\_\_\_\_ and in case of variable is \_\_\_\_.

- ▶ Call by value, call by reference
- ▶ Call by referene, call by reference
- ▶ **Call by reference, call by value**
- ▶ Call by value, call by value

Question No: 11 ( M - 1 ) .  
What does STL stand for?

- ▶ Source template library
- ▶ **Standard template library**
- ▶ Stream template library
- ▶ Standard temporary library

Question No: 12 ( M - 1 ) .  
Skill(s) that is/are needed by programmers \_\_\_\_\_.

- ▶ Paying attention to detail
- ▶ Think about the reusability
- ▶ Think about user interface
- ▶ **All of the given options**

Question No: 13 ( M - 1 ) .  
For which array, the size of the array should be one more than the number of elements in an array?

- ▶ int
- ▶ double
- ▶ float
- ▶ **char**

Question No: 14 ( M - 1 ) .  
new and delete are \_\_\_\_\_ whereas malloc and free are \_\_\_\_\_.

- ▶ Functions, operators
- ▶ Classes, operators
- ▶ **Operators, functions**
- ▶ Operators, classes

Question No: 15 ( M - 1 ) .  
Friend functions are \_\_\_\_\_ of a class.

- ▶ Member functions
- ▶ Public member functions
- ▶ Private member functions
- ▶ **Non-member functions**

Question No: 16 ( M - 1 ) .  
The prototype of friend functions must be written \_\_\_\_\_ the class and its definition must be written \_\_\_\_\_.

- ▶ inside, inside the class
- ▶ **inside, outside the class**
- ▶ outside, inside the class
- ▶ outside, outside the class

Question No: 17 ( M - 1 ) .  
If overloaded plus operator is implemented as non-member function then which of the following statement will be true for the statement given below?

obj3 = obj1 + obj2 ;

- ▶ obj2 will be passed as an argument to + operator whereas obj2 will drive the + operator
- ▶ **obj1 will drive the + operator whereas obj2 will be passed as an argument to + operator**
- ▶ Both objects (obj1, obj2) will be passed as arguments to the + operator
- ▶ Any of the objects (obj1, obj2) can drive the + operator

Question No: 18 ( M - 1 ) .  
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: 19 ( M - 1 ) .  
For cin, the source is normally a \_\_\_\_\_ and destination can be \_\_\_\_\_.

- ▶ File, native data type
- ▶ **Disk, user-define type**

- ▶ **Keyboard, variable**
- ▶ File, user-define type

Question No: 20 ( M - 1 ) .

We can also do conditional compilation with preprocessor directives.

- ▶ **True**
- ▶ False

Question No: 21 ( M - 1 ) .

The programs, in which we allocate static memory, run essentially on \_\_\_\_\_

- ▶ Heap
- ▶ System Cache
- ▶ None of the given options
- ▶ **Stack**

Question No: 22 ( M - 1 ) .

The default value of a parameter can be provided inside the \_\_\_\_\_

- ▶ function prototype
- ▶ function definition
- ▶ **both function prototype or function definition**
- ▶ none of the given options.

Question No: 23 ( M - 1 ) .

While calling function, the arguments are assigned to the parameters from \_\_\_\_\_.

- ▶ **left to right.**
- ▶ right to left
- ▶ no specific order is followed
- ▶ none of the given options.

Question No: 24 ( M - 1 ) .

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**
- ▶ Two
- ▶ N arguments

Question No: 25 ( M - 1 ) .

With user-defined data type variables (Objects), self assignment can produce \_\_\_\_\_.

- ▶ Syntax error
- ▶ **Logical error**
- ▶ Link error
- ▶ Non of the given options

Question No: 26 ( M - 1 ) .

Assignment operator is used to initialize a newly declared object from existing object.

- ▶ **True**
- ▶ False

Question No: 27 ( M - 1 ) .

When an object of a class is defined inside an other class then,

- ▶ Constructor of enclosing class will be called first
- ▶ **Constructor of inner object will be called first**
- ▶ Constructor and Destructor will be called simultaneously
- ▶ None of the given options

Question No: 28 ( M - 1 ) .

In the member initializer list, the data members are initialized,

- ▶ From left to right
- ▶ From right to left
- ▶ **In the order in which they are defined within class**
- ▶ None of the given options

Question No: 29 ( M - 1 ) .  
new operator allocates memory from free store and return \_\_\_\_\_.

- ▶ **A pointer**
- ▶ A reference
- ▶ An integer
- ▶ A float

Question No: 30 ( M - 1 ) .  
"new" and "delete" keywords are \_\_\_\_\_ in C++ language.

- ▶ Built-in- Function
- ▶ **Operators**
- ▶ Memory Allocation Function
- ▶ None of the given options

Operator overloading can be performed through \_\_\_\_\_.

- ▶ Classes
- ▶ **Functions**
- ▶ Operators
- ▶ Reference

**Question No: 2 ( M - 1 ) .**

When a value is referred by a normal variable then it is known as,

- ▶ **Direct Reference**
- ▶ Indirect Reference
- ▶ Partial Reference
- ▶ Proper Reference

When a value is referred by a normal variable is known as direct reference

**Question No: 3 ( M - 1 ) .**

Which of the following function is used to increase the size of already allocated memory chunk?

- ▶ malloc
- ▶ calloc
- ▶ **realloc**
- ▶ free

(FQ, vuzs, 2010)

**Question No: 4 ( M - 1 ) .**

Which of the following is NOT a preprocessor directive?

- ▶ **#error**

- ▶ #define
- ▶ #line
- ▶ #undef

### list of preprocessors

• #include • #include "filename" • #define • #undef • #ifdef • #ifndef • #if • #else • #elif • #endif • #error • #line • #pragma • #assert

### Question No: 5 ( M - 1 ) .

The stream objects **cin** and **cout** are included in which header file?

- ▶ **iostream.h**
- ▶ fstream.h
- ▶ istream.h
- ▶ ostream.h

<http://www.vuzs.info/>

### Question No: 6 ( M - 1 ) .

Overloaded delete operator function takes the same parameter as an argument returned by new operator function.

- ▶ **True**
- ▶ False

The same pointer that is returned by the new operator, is passed as an argument to the delete operator. These rules apply to both, if operators (new and delete) are overloaded as member or non-member operators (as global operators).

### Question No: 7 ( M - 1 ) .

When an array of object is created dynamically then there is no way to provide parameterized constructors for array of objects.

- ▶ **True**
- ▶ False

if we are allocating an array of objects, there is no way to pass arguments to objects' constructors. Therefore it is required that the objects that are stored in such an array have a no-argument constructor.

### Question No: 8 ( M - 1 ) .

C is widely known as development language of \_\_\_\_\_ operating system.

- ▶ Linux
- ▶ Windows
- ▶ **Unix**
- ▶ Mac OS

In the start C became widely known as the development language of the UNIX operating system, and the UNIX operating system was written by using this C language. The C language is so powerful that the compiler of C and other various operating systems are written in C.

<http://vuzs.net>

### Question No: 9 ( M - 1 ) .

Computer can understand only machine language code.

- ▶ **True**
- ▶ False

### Question No: 10 ( M - 1 ) .

We can not define a function as a friend of a Template class.

- ▶ True
- ▶ **False**

Class templates can have **friends**. A class or class template, function, or function template can be a friend to a template class. Friends can also be specializations of a class template or function template, but not partial specializations.

**Question No: 11 ( M - 1 ) .**

**What will be the value of 'a' and 'b' after executing the following statements?**

a = 3;

b = a++;

▶ 3, 4

▶ 4, 4

▶ 3, 3

▶ **4, 3**

**Question No: 12 ( M - 1 ) .**

Consider the following code segment. What will be the output of following code?

```
int addValue (int *a){
```

```
int b = (*a) + 2;
```

```
return b ;
```

```
}
```

```
main () {
```

```
int x =6 ;
```

```
cout << x << "," ;
```

```
cout << addValue(&x) << "," ;
```

```
cout << x ;
```

```
}
```

<http://vuzs.net>

▶ **6,8,6**

▶ 6,6,8

▶ 6,8,8

▶ 6,6,6

<http://www.vuzs.info/>

**Question No: 13 ( M - 1 ) .**

\_\_\_\_\_ is used to trace the logic of the program and correct the logical errors.

▶ Compiler

▶ Editor

▶ Linker

▶ **Debugger**

**Question No: 14 ( M - 1 ) .**

new and delete are \_\_\_\_\_ whereas malloc and free are \_\_\_\_\_.

▶ Functions, operators

▶ Classes, operators

▶ **Operators, functions**

▶ Operators, classes

Hence, we can call new and delete operators, P# 342

we have allocated a memory space for our use by malloc function. P# 285

**Question No: 15 ( M - 1 ) .**

Like member functions, \_\_\_\_\_ can also access the private data members of a class.

▶ Non-member functions

▶ **Friend functions**

▶ Any function outside class

▶ None of the given options

**Question No: 16 ( M - 1 ) .**

Which situation would require the use of a non-member overloaded operator?

▶ The overloaded operator is an Assignment operator.

<http://vuzs.ne>

- ▶ The left most operand is an object of a class.
- ▶ **The left operand is built-in data type.**
- ▶ The operator returns a reference.

(100% confirmed by Rainbowbright)

**Question No: 17 ( M - 1 ) .**

The stream insertion and stream extraction operators are already overloaded for \_\_\_\_\_.

- ▶ User-defined data types
- ▶ **Built-in data types**
- ▶ User-defined and built-in data types
- ▶ None of the given options

**Question No: 18 ( M - 1 ) .**

If we define an identifier with the statement **#define PI 3.1415926** then during the execution of the program the value of PI \_\_\_\_\_.

- ▶ can not be replaced
- ▶ None of the given options
- ▶ **Remain constant.**
- ▶ can be changed by some operation

**Question No: 19 ( M - 1 ) . vuzs**

Assignment operator is -----associative.

- ▶ **right**
- ▶ left
- ▶ binary
- ▶ unary

You can assign values to several variables in a single statement. For example, the following code sets the contents of apples and oranges to the same value:

apples = oranges = 10;

The assignment operator is right associative, so this statement executes by first storing the value 10 in oranges and then storing the value in oranges in apples, so it is effectively

apples = (oranges = 10);

<http://www.vuzs.info/>

**Question No: 20 ( M - 1 ) .**

When ever dynamic memory allocation is made in C/C++, it is freed\_\_\_\_\_.

- ▶ **Explicitly**
- ▶ Implicitly
- ▶ Both explicitly and implicitly
- ▶ None of the given options

(Rainbowbright, vuzs. jul2011)

**Question No: 21 ( M - 1 ) .**

The appropriate data type to store the number of rows and columns of the matrix is\_\_\_\_\_.

- ▶ float
- ▶ **int**
- ▶ char
- ▶ none of the given options.

**Question No: 22 ( M - 1 ) .**

Which of the following function do NOT initialize the chunk of memory to all zero?

- ▶ calloc() function
- ▶ Both malloc() and calloc()
- ▶ None of the above
- ▶ **malloc() function**

The **malloc** function differs from **calloc** in the way that the space allocated by **malloc** is not initialized and contains any values initially.

**Question No: 23 ( M - 1 ) .**

The function free() returns back the allocated memory got through calloc and malloc to \_\_\_\_\_ .

- ▶ stack
- ▶ **heap**
- ▶ stack and heap
- ▶ None of the given options

<http://vuzs.net>

**Question No: 24 ( M - 1 ) .**

width() is member function of \_\_\_\_\_

- ▶ cin object
- ▶ cout object
- ▶ **Both cin and cout object**
- ▶ None of the given option

**Question No: 25 ( M - 1 ) .**

Templates are not type safe.

- ▶ true
- ▶ **false**

Templates are type-safe. This is because the types that templates act upon are known at compile time, so the compiler can perform type checking before errors occur.

**Question No: 26 ( M - 1 ) .**

A Matrix can be composed of ints, floats or doubles as their elements. Best way is to handle this ,

- ▶ Write a separate class to handle each
- ▶ **Use templates**
- ▶ Use strings to store all types
- ▶ None of the given options

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

**Left**

**2. We cannot increment \_\_\_\_\_.**

**references**

**3. We can \_\_\_\_\_ pointer.**

**all of the given**

**4. We can \_\_\_\_\_ references.**

**None of the given**

5. What will be the correct syntax for the following function call?

`float add (int &);`

`add(x);`

6. An instance of a class is called \_\_\_\_\_.

`object`

7. The \_\_\_\_\_ is called automatically when an object destroys

`destructor`

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

1. `deallocate memory`

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

`private`

10. Member functions of the class \_\_\_\_\_ main program.

1. `are accessible from`

11. Overloading means :

1. `Using the same name to perform multiple tasks or different tasks depending on the situation.`

12. The main advantage of function overloading is \_\_\_\_\_.

1. `The program becomes more readable`

13. You cannot overload the \_\_\_\_\_ operator.

1. `? :`

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

1. efficiency

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

1. 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'

17. The dynamic memory allocation uses \_\_\_\_\_ whereas static memory allocation uses \_\_\_\_\_.

1. heap , stack

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

```
#include <iostream>
#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=24 y=47

19. NULL has been defined in \_\_\_\_\_ header file.

1. Stdlib.h

20. Symbolic constant PI can be defined as:

1. #define PI 3.14

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

1. to the private data members

22. C++ was developed by \_\_\_\_\_.

1. BejarneStroustrup

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

1. static variables

24. Encapsulation means \_\_\_\_\_.

1. that the data of a class can be accessed from outside

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

1. 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. **getche()** 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. iomanip

**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
4. None of the given

**70. The memory allocation in C++ is carried out with the help of \_\_\_\_\_.**

1. NULL pointer
2. new operator
3. dot operator
4. + operator

**71. If B is designated as friend of A, B can access A's non-public members.**

1. B cannot access private member of A
2. B cannot access protected member of A
3. A can access non-public members of B
4. A cannot access B

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

1. the new operator returns 2
2. the new operator returns 1
3. the operator returns 0
4. free operator returns nothing

**73. 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.**

1. upper case alphabets
2. lower case alphabets
3. double underscore
4. None of the given

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

1. in one direction only
2. in two directions
3. to all classes
4. to the data members of the friend class only

**75. The malloc function takes \_\_\_\_\_ argument(s).**

1. two
2. three
3. four
4. one

**76. The constructor contains \_\_\_\_\_.**

1. return type
2. no return type
3. objects
4. classes

**77. What will be the output of the following c++ code?**

```
#include<iostream.h>
#define max 100
main()
{
#ifdef max
Cout<<"Hellow;
}
```

1. Hello
2. "Hellow"
3. Max is 100
4. Error

**78. Once we have defined a symbolic constant value using #define, that value \_\_\_\_\_ during program execution**

1. can be changed
2. cannot be changed
3. varies
4. becomes zero

**79. The memory allocation functions return a chunk of memory with a pointer of type \_\_\_\_\_.**

1. integer
2. float
3. ptr
4. void

**80. A class can be declared as a \_\_\_\_\_ of other class.**

1. member
2. member function
3. friend

4. part

**81. To avoid dangling reference, don't return \_\_\_\_\_.**

1. the reference of a local variable from the function
2. the reference of a global variable from the function
3. the reference of a static variable from the function
4. the reference of a private data member from the function

**82. Constructor is itself a \_\_\_\_\_ of C++ and \_\_\_\_\_.**

1. class, can be overloaded
2. function, cannot be overloaded
3. function, can be overloaded
4. object, can not be initialized

**83. The parameter passed to isdigit() function is \_\_\_\_\_ variable.**

1. Character
2. Boolean
3. Integer
4. Float

**84. char \*\*argv can be read as \_\_\_\_\_.**

1. pointer to pointer
2. pointer to char
3. pointer to pointer to char
4. None of the given

**85. To read command-line arguments, the main() function itself must be given \_\_\_\_\_ arguments.**

1. 1
2. 2
3. 3
4. 4

**86. How many bytes an integer type pointer intPtr will jump in memory if the statement below is executed?  
intPtr += 2 ;**

1. 2
2. 4
3. 8
4. 12

**87. The increment of a pointer depends on its \_\_\_\_\_.**

1. variable
2. value
3. data type
4. None of the given

**88. The statement `cout<<yptr` will show the \_\_\_\_\_ theyptr points to.**

1. Value
2. memory address
3. variable
4. None of the given

**89. \_\_\_\_\_ is used as a dereferencing operator.**

1. \*
2. +
3. -
4. None of the above

**90. Transpose of a matrix means that when we interchange rows and columns \_\_\_\_\_.**

1. the first row becomes the Last column
2. the first row becomes the first column
3. the Last row becomes the first column
4. the first column becomes the first row

**91. Individual characters in a string stored in an array can be accessed directly using array \_\_\_\_\_.**

1. superscript
2. script
3. subscript

4. value

**92. We can define a matrix as \_\_\_\_\_ array.**

1. Sorted
2. Unsorted
3. Single dimensional
4. Multi dimensional

**93. A \_\_\_\_\_ is an array of characters that can store number of character specified.**

1. Char
2. String
3. Multidimensional array
4. Data type

**94. Given a two dimensional array of integers, what would be the correct way of assigning the value 6 to the element at third row and fourth column?**

1. `array[3][4] = 6 ;`
2. `array[2][4] = 6 ;`
3. `array[4][3] = 6 ;`
4. `array[2][3] = 6 ;`

**95. \_\_\_\_\_ of a variable means the locations within a program from where it can be accessed.**

1. Data type
2. Visibility
3. Value
4. Reference

**96. Which of the following function call is "call by reference" for the following function prototype?  
`int add (int *);`**

1. `add(&x);`
2. `add(int x);`
3. `add(x);`

4. add(\*x);

**97. Which of the following function call is "call by reference" for the following function prototype?  
float add (float \*);**

1. add(&x);
2. add(float x);
3. add(x);
4. add(\*x);

**98. Which of the function call is call by value for the following function prototype?  
float add(float);**

1. add(&x);
2. add(x);
3. add(float x);
4. add(\*x);

**99. Which of the function call is "call by value" for the following function prototype?  
float add(int);**

1. add(&x);
2. add(x);
3. add(int x);
4. add(\*x);

**100. Return type of a function that does not return any value must be \_\_\_\_\_.**

1. char
2. int
3. void
4. double

**101. \_\_\_\_\_ will be used for enclosing function statements into a block.**

1. " "

2. ()

3. []

4. {}

**102. What is the output of the following code if the 2nd case is true**

```
switch (var) {
case 'a': cout<<"apple"<<endl;
case 'b':cout<<"banana"<<endl;
case 'm':cout<<"mango"<<endl;
default: cout<<"any fruit"<<endl;
}
```

1. banana

2. banana  
any fruit

3. banana  
mango  
any fruit

4. None of the given

**103. When the break statement is encountered in a loop's body, it transfers the control \_\_\_\_\_ from the current loop.**

1. Inside

2. Outside

3. To break statement

4. To continue statement

**104. What is the output of the following code if the 3rd case is true**

```
switch (var) {
case 'a':cout<<"apple"<<endl;
case 'b':cout<<"banana"<<endl;
case 'm':cout<<"mango"<<endl;
default: cout<<"any fruit"<<endl;
}
```

1. mango
2. mango  
any fruit
3. apple
4. None of the given

**105. What is the output of the following code, if the first case is true**

```
switch (var) {
case 'a':cout<<"apple"<<endl;
case 'b':cout<<"banana"<<endl;
case 'm':cout<<"mango"<<endl;
default: cout<<"any fruit"<<endl;
}
```

1. apple
2. apple  
any fruit
3. apple  
banana  
mango  
any fruit
4. none of above

**106. What will be the output of following code segment?**

```
for (int i = 2; i<10; i++) {
if (i == 5) continue;
cout<<i<< ", " ;
}
```

1. 2,3,7,8,9
2. 2,3,4,6,7,8,9
3. 2,3,4
4. 4,6,7,8,9

**107. \_\_\_\_\_ statement is used to terminate the processing of a particular case and exit from switch structure.**

1. if
2. goto
3. break
4. continue

**108. What will be the result of the expression  $j = i++$ ; if initially  $j = 0$  and  $i = 5$ ?**

1. 0
2. 5
3. 6
4. 4

**109. What will be the result of the expression  $k = ++m$ ; if initially  $k = 0$  and  $m = 4$ ?**

1. 0
2. 5
3. 6
4. 4

**110. What will be the result of the expression  $k = ++m$ ; if initially  $k = 0$  and  $m = 5$ ?**

1. 0
2. 5
3. 6
4. 4

**111. How many times the following do-while loop will execute?**

```
int k = 10; do { cout << "Statements" << endl; k -= 2; }
while(k > 0);
```

1. 4
2. 5
3. 6
4. 7

**112. Which of the following loops checks the test condition at the end of the loop?**

1. While
2. Do-While
3. For
4. Nested Loop

**113. The operators ++ and -- are used to increment or decrement the value of a variable by \_\_\_\_\_.**

1. 3
2. 2
3. 1
4. 4

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

1. 0
2. 1
3. 2
4. 3

**115. A \_\_\_\_\_ structure specifies that an action is to be repeated while some condition remains true.**

1. Control
2. Logical
3. Repetition
4. Relational

**116. !( x > 3) means in C++ that**

1. x is greater than 3
2. x is less than or equal to 3
3. x is less than 3
4. x is equal to 3

**117. When the logical operator && combines two expressions then the result will be true only when the both expressions are \_\_\_\_\_.**

1. Logical
2. Arithmetic
3. true
4. false

**118. < and > both are \_\_\_\_\_ operators.**

1. Arithmetic
2. Relational
3. Logical
4. Mathematical

**119. What will be the value of variable "input" if the initial value of input is 67?**

```
if(input >= 50)
input = input + 1;
if(input <= 75)
input = input + 2;
else
input = input - 1;
```

1. 68
2. 69
3. 70
4. 66

**120. !( x< 3) means in C++ that**

1. x is less than 3
2. x is greater than or equal to 3
3. x is greater than 3
4. x is equal to 3

**121. != operator is used to check whether the operand on the left-hand-side is \_\_\_\_\_ to the operand on the right-hand-side.**

1. Less than or equal
2. Greater than or equal
3. Not equal

4. Approximately equal to

**122. When the if statement consists more than one statement then enclosing these statement in curly braces is,**

1. Not required
2. Good programming
3. Relevant
4. **Must**

**123. The most suitable data type for number 325.25 is**

\_\_\_\_\_.

1. char
2. int
3. short
4. **float**

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

1. 10
2. 40.5
3. **42**
4. 41

**125. Which of the following will be the most appropriate data type to store the value 63.547?**

1. Integer
2. Character
3. Short
4. **Float**

**126. In the given expression which operator will be evaluated first?  $10 + (6 / 2) - 2 * 3$ ?**

1. +
2. -
3. **/**

4. \*

**127. What will be the value of the variable output in the given piece of code?**

```
double output = 0;
output = (2 + 2) * 4 + 2 / (4 - 2);
```

1. 15

2. 17

3. 12

4. 11

**128. It is the job of \_\_\_\_\_ to transfer the executable code from hard disk to main memory.**

1. interpreter

2. Debugger

3. Linker

4. Loader

**129. In computer systems there are mainly \_\_\_\_\_ type of softwares.**

1. 1

2. 2

3. 3

4. 4

**130. \_\_\_\_\_ will explain the function of a program.**

1. Comments

2. Debugger

3. Compiler

4. Linker

**131. if (a>b && a>c) then the condition will be true only if**

1. Both a>b and a>c are true

2. a>b is false and a>c is true

3. a>b is true and a>c is false

4. Both  $a > b$  and  $a > c$  are false

**132. A variable of character data type occupies \_\_\_\_\_ byte(s) in memory.**

1. 1

2. 2

3. 4

4. 8

**133. We must include the header file \_\_\_\_\_ to convert the value of one type into another type using built-in functions.**

1. conio.h

2. stdlib.h

3. string.h

4. iostream.h

**134. A function is a block of statements that can be defined once and used \_\_\_\_\_ in the program.**

1. One time

2. Two times

3. Three times

4. As many times as user wants

**135. Select the correct way to assign the address of first element of array to pointer?**

1. `int *ptr = &data[1];`

2. `int *ptr = &data;`

3. `int *ptr = data;`

4. `int *ptr = data[0];`

**136. Consider the following code segment. What will be the output of following code?**

```
int addValue (int *a){
 int b = (*a) + 2;
 return b;
}
```

```
main() {
 int x = 6;
 cout<<addValue(&x)<<" ";
 cout<<x;
}
```

1. 6,8,6

2. 6,6,8

3. 6,8,8

4. 6,6,6

**137. Here the code is given below. You have to identify the problem in the code.**

```
while(i< 10) && (i> 24))
```

1. the logical operator && cannot be used in test condition

2. the while loop is an exit-condition loop

3. the test condition is always true

4. the test condition is always false

**138. The correct syntax of do-while loop is \_\_\_\_\_.**

1. (condition) while; do {statements};

2. {statements;} do-while();

3. while(condition); do {statements};

4. do {statements;} while (condition);

**139. Matrix is defined as \_\_\_\_\_.**

1. Single dimensional array

2. Multi-dimensional array

3. Vector product

4. Scalar product

**140. In programming, comments are used to explain the functioning of the \_\_\_\_\_.**

1. Debugger

2. Editor

3. Program

4. Linker

**141. Operating System is a type of a/an \_\_\_\_\_.**

1. application software
2. system software
3. computer language
4. interpreter

**142. From the options given, you need to choose the option which is true for the given code.**

```
for (int i = 1; i>0; i++) {
/*loop code*/
}
```

1. the logical operator && cannot be used in a test condition
2. the while loop is an exit-condition loop
3. the test condition is always false
4. the test condition is always true

**143. Which of the following values are used in C/C++ to represent true and false?**

1. 1 and 0
2. 1 and -1
3. 11 and 00
4. any numerical value

**144. 'While' loop may execute \_\_\_\_\_ or more times.**

1. three
2. zero
3. two
4. one

**145. Body of any function is enclosed within \_\_\_\_\_.**

1. { }
2. ( )
3. [ ]

4. " "

**146. What will be the correct syntax for initialization of a pointer ptr with string "programming"?**

1. char ptr = 'programming';
2. char \*ptr = "programming";
3. char \*ptr = 'programming';
4. \*ptr = "programming";

**147. Which one of the given option is not a mode for reading/writing the data from a file?**

1. in
2. out
3. trunc
4. get

**148. Which of the following operators is used to access the value of variable pointed by a pointer?**

1. \* operator
2. -> operator
3. && operator
4. &operator

**149. In case of single dereferencing, the value of the \_\_\_\_\_ is the address of the \_\_\_\_\_.**

1. pointer, variable
2. pointer, constant
3. variable, pointer
4. constant, pointer

**150. The remainder (%) operator is a \_\_\_\_\_ operator.**

1. Logical
2. Arithmetic
3. Relational
4. Conditional

**151. What will be the output of following code?**

```
int x = 10;
cout<<"x="<<x;
```

1. 10
2. "x=10"
3. x=10
4. 10=x

**152. The purpose of using cout<< is to \_\_\_\_\_.**

1. Display information on the screen
2. Read the data from keyboard
3. Read the data from a file
4. Write into a file

**153. Which of the following data types will be assumed if no data type is specified with constant?**

1. short
2. float
3. int
4. double

**154. When an array element is passed to a function, it is passed by \_\_\_\_\_.**

1. reference
2. data type
3. value
4. data

**155. While programming, it is good to provide an easy to understand and easy to use interface; this programming skill is called \_\_\_\_\_.**

1. scalability
2. usability
3. reliability
4. sustainability

**156. \_\_\_\_\_ executes all the lines before error and stops at the line which contains the error.**

1. Intrepreter

2. Compiler

3. Linker

4. Debugger

**157. Which of the following is the correct syntax to access the value of first element of an array using pointer ptr?**

1. ptr[0]

2. \*(ptr+1)

3. ptr[1]

4. \*ptr[0]

**158. C is a/an \_\_\_\_\_ language.**

1. low level

2. object based

3. object oriented

4. function oriented

**159. \_\_\_\_\_ of a function is also known as signature of a function.**

1. Definition

2. Declaration

3. Calling

4. Invoking

**160. \_\_\_\_\_ are very good tools for code reuse.**

1. operators

2. loops

3. functions

4. variables

**161. If any break statement is missed in switch statement then \_\_\_\_\_.**

1. compiler will give error
2. this may cause a logical error
3. no effect on program
4. program stops its execution

**162. A 2D array multi[5][10] can be accessed using the array name as \*\*multi, this technique is called \_\_\_\_\_.**

1. Single referencing
2. Single dereferencing
3. Double referencing
4. Double dereferencing

**163. In C/C++, the default command line arguments passed to the main function are \_\_\_\_\_.**

1. float argc, char \*\*argv
2. int argc, char \*\*argv
3. int \*argc, char \*argv
4. int argc, float \*\*argv

**164. A record is a group of related \_\_\_\_\_.**

1. Data
2. Fields
3. Bytes
4. Files

**165. The microsoft word document (.doc) is a kind of \_\_\_\_\_.**

1. Sequential File
2. Random AccessFile
3. Binary Access File
4. Executable File

**166. NULL character is used to indicate the \_\_\_\_\_ of string.**

1. Start
2. End

- 3. Begin
- 4. Middle

**167. How many dimensions does n-dimensional array has?**

- 1. n dimensions
- 2.  $2n$  dimensions
- 3.  $(n+1)$  dimensions
- 4.  $(n-1)$  dimensions

**168. Which of the following function call is "call by reference" for the following function prototype?**

- 1. `func(int &num);`
- 2. `func(&num);`
- 3. `func(*num);`
- 4. `func(num);`

**169. The loop which is most suitable to be used when the number of iterations is known is called \_\_\_\_\_.**

- 1. for
- 2. while
- 3. do-while
- 4. all looping processes require that the iterations be known.

**170. In C/C++, the string constant is enclosed in \_\_\_\_\_.**

- 1. curly braces { }
- 2. parentheses ( )
- 3. single quotes ' '
- 4. double quotes " "

**171. In order to get the right most digit of a number, we divide this number by 10 and take \_\_\_\_\_.**

- 1. Its remainder
- 2. Its quotient
- 3. Its divisor
- 4. The number

**172. What is the correct syntax to declare an array of size 10 of int data type?**

1. `int [10] name;`
2. `name[10] int;`
3. `int name[10];`
4. `int name[];`

**173. How many bytes of memory are occupied by array 'str'?**

`char str[] = "programming";`

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

**174. Suppose that an integer type pointer contains a memory address 0x22f230. What will be the new memory address if we increment this pointer by one?**

1. 0x22f231
2. 0x22f234
3. 0x22f226
4. 0x22f238

**175. Which of the following if missing would result in infinite recursion in case of recursive function?**

1. Recursive call
2. Base case
3. Function parameters
4. Local variables

**176. Whenever we use a library function or a predefined object or macro, we need to use a \_\_\_\_\_.**

1. source file
2. object file
3. header file
4. exe file

**177. Switch statement deals with \_\_\_\_\_ type of data.**

1. Integer
2. Float
3. Character
4. Both Integer and Character

**178. Both compiler and \_\_\_\_\_ are used to translate program into machine language code.**

1. debugger
2. linker
3. loader
4. interpreter

**179. TWAIN stands for \_\_\_\_\_.**

1. Technology With An Interesting Name
2. Technology Without An Informative Name
3. Technology Without An Interesting Name
4. Technology With An Informative Name

**180. The parameter passed to isdigit() function is \_\_\_\_\_.**

1. a character variable
2. a boolean variable
3. an integer variable
4. a character string

**181. C++ views each file as a sequential stream of \_\_\_\_\_.**

1. Bits
2. Bytes
3. Numbers
4. Words

**182. Structure is a collection of \_\_\_\_\_ under a single name.**

1. only functions

2. only variables

3. both functions and variables

4. only data types

**183. The default mode for writing into a file using ofstream object is \_\_\_\_\_.**

1. out

2. bin

3. app

4. ate

**184. The memory address of the first element of an array is called \_\_\_\_\_.**

1. floor address

2. foundation address

3. first address

4. base address

**185. We want to access array in random order which of the following approach is better?**

1. Pointer

2. Array index

3. Both pointers and array index are better

4. Matrix

**186. The \_\_\_\_\_ structure is a multiple-selection construct which makes the code more efficient and easy to read and understand.**

1. multiple-if

2. switch

3. if-else

4. else-if

**187. Which of the following is not a reserved word in C/C++?**

1. int

2. float

3. double

4. sum

**188. To access rand(), which library is required to be included in program?**

1. conio.h

2. stdio.h

3. stdlib.h

4. iostream.h

**189. What is the highest legal index for the following array?**  
**int arr[4]**

1. 4

2. 3

3. 2

4. 1

**190. Word processor is a type of a/an \_\_\_\_\_.**

1. operating system

2. application software

3. device driver

4. utility software

**191. Identify the correct option which is used for calling the function float area (int).**

1. area(&num);

2. area(num);

3. area(int num);

4. area(\*num);

**192. The \_\_\_\_\_ statement allows us to select from multiple choices based on a set of fixed values for a given expression.**

1. switch

2. break

3. continue

4. goto

**193. C is widely known as development language of \_\_\_\_\_ operating system.**

1. Windows

2. Unix

3. Mac OS

4. Linux

**194. To convert the value of one type into another type using built-in functions, we include \_\_\_\_\_ header file.**

1. conio.h

2. stdlib.h

3. iostream.h

4. string.h

**195. The keyword \_\_\_\_\_ is used to get some value back from a function.**

1. return

2. break

3. continue

4. goto

**196. The function seekg() takes \_\_\_\_\_ parameter(s).**

1. 0

2. 1

3. 2

4. 3

**197. The function write() takes \_\_\_\_\_ as parameter(s).**

1. String of pointer type

2. String of variable lengths, no. of bytes to be read and flags

3. Pointer array of characters and a delimiter

4. String and no. of bytes to be written

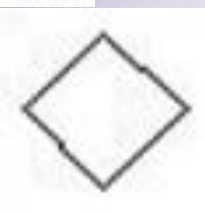
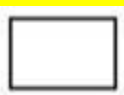
**198. When the logical operator AND (&&) combines two expressions exp1 and exp2 then the result will be true only \_\_\_\_\_.**

1. When both exp1 and exp2 are true
2. When both exp1 and exp2 are false
3. When exp1 is true and exp2 is false
4. When exp1 is false and exp2 is true

**199. Syntax of a union is identical to \_\_\_\_\_.**

1. structure
2. class
3. function
4. loop

**200. Which one of the symbol is used to represent a decision in a flow chart?**

1. 
2.  correct
3. 
4. 

**201. In Flow Chart, flow of control is represented by \_\_\_\_\_.**

1. Rectangle
2. Circle
3. Diamond
4. Arrow

**202. There can be \_\_\_\_\_ 'default' statement(s) in any switch structure.**

1. 1

2. 2

3. 3

4. n

**203. The condition in loop should be a(n) \_\_\_\_\_.**

1. Constant Expression

2. Boolean Expression

3. Primary Expression

4. Arithmetic Expression

**204. How many nested loops would be required to manipulate n-dimensional array?**

1. n

2.  $n + 1$

3.  $n - 1$

4.  $2n$

**205. Which of the following is not an example of int data type?**

1. 0

2. -32

3. 65531

4. -4

**206. We should use \_\_\_\_\_ for clarity and to force the order of evaluation in an expression.**

1. brackets []

2. parenthesis ()

3. curly braces {}

4. quotation marks " "

**207. Which of the following is the starting index of an array in C++?**

1. 0

2. 1

3. -1
4. 2

**208. The statement  $x += y$  can be interpreted as \_\_\_\_\_.**

1. Adding the value of the x to the value of the y and storing the result in x
2. Adding the value of the y to the value of x, store the result in y
3. Adding the value of the x to the value of x, store the result in x
4. Adding the value of the y to the value of y, store the result in x

**209. Given a 2D array of integers, what would be the correct way of assigning the value 5 to the element at second row and third column?**

1.  $m[2][3] = 5;$
2.  $m[3][2] = 5;$
3.  $m[1][2] = 5;$
4.  $m[2][3] = '5';$

**210. Array is a data structure that stores \_\_\_\_\_.**

1. Memory addresses
2. Variables
3. Data type
4. Data

**211. A program statement that invokes a function is called \_\_\_\_\_.**

1. function declaration
2. function call
3. function definition
4. function prototype

**212. If a function has been declared but not defined before its function call then it is termed as \_\_\_\_\_.**

1. logical error
2. syntax error
3. run time error
4. program time error

**213. The compiler of C language is written in \_\_\_\_\_ language.**

1. JAVA
2. BASIC
3. FORTRAN
4. C

**214. Which one of the below functions is not included in ctype.h header file?**

1. isdigit(int c)
2. isxdigit(int c)
3. tolower(int c)
4. getdigit(int c)

**215. Which function is used to locate the first occurrence of a character in any string?**

1. strchr()
2. strstr()
3. strtok()
4. strlen()

**216. To access the data members of structure, \_\_\_\_\_ is used.**

1. Logical operator
2. Dereference operator
3. Dot operator
4. Address operator

**217. In the following nested For Loop, which loop will run most number of times?**

```
for(inti = 0; i < 5; i++)
{
```

```
for(int k = 0; k < 5; k++)
{

}
```

1. Outer loop
2. Inner loop
3. Both loops run equal number of times
4. Depends upon the statements in the inner loop's body

**218. Structure use \_\_\_\_\_ allocation.**

1. Queue
2. Heap
3. Cache
4. Stack

**219. \_\_\_\_\_ function give the position of the next character to be read from that file.**

1. tellp()
2. tellg()
3. seekg()
4. seekp()

**220. What will be the size of the following character array?  
char name[] = "Adeel";**

1. 5
2. 6
3. 4
4. 7

**221. Function prototype is written,**

1. Within main function
2. After the return statement in main
3. Before the return statement in main
4. Before call of that function

**222. Which one of the following languages has been used to write the compiler of "C" language?**

- 1. Java
- 2. Fortran
- 3. Basic
- 4. C

**223. A hierarchy of classes which are used to deal with console and disk files are called \_\_\_\_\_.**

- 1. Stream classes
- 2. Simple classes
- 3. Binary classes
- 4. IO classes

**224. \_\_\_\_\_ stops execution at the line that contains error(s) in the code.**

- 1. Compiler
- 2. Debugger
- 3. Interpreter
- 4. Linker

**225. C++ is a \_\_\_\_\_ language.**

- 1. High level
- 2. Low level
- 3. Machine
- 4. Assembly language

**226. How many elements are stored in the following?  
int matrix [4][5];**

- 1. 9
- 2. 20
- 3. 25
- 4. 10

**227. \_\_\_\_\_ is a substitute of multiple if statement.**

1. if. . .elseif statement

2. Continue statement

3. Break statement

4. Default statement

**228. if**

**int sum = 54;**

**Then the value of the following statement is**

**sum = sum - 3;**

1. 52

2. 50

3. 51

4. 57

**229. What will be the correct syntax for declaration of the following statement?**

**"ptr is a constant pointer to an integer"**

1. const \* int myptr;

2. const int \*myptr;

3. int const \*ptr;

4. int \*const ptr;

**230. \_\_\_\_\_ operator is used to pass the address of a variable in call by reference method.**

1. %

2. +

3. @

4. &

**231. \_\_\_\_\_ data type can operate on modulus operator.**

1. int

2. float

3. char

4. double

**232. Whenever some number is added in an array name, it will jump as many \_\_\_\_\_ as the added number.**

1. rows

2. value

3. column

4. bytes

**233. Suppose that an integer type pointer contains a memory address 0x22f220. What will be the new memory address if we increment this pointer by one?**

1. 0x22f221

2. 0x22f222

3. 0x22f223

4. 0x22f224

**234. \_\_\_\_\_ is the pointer which determines the position in a file from where the next read operation occurs.**

1. put

2. seek

3. get

4. tell

To access the data members of structure \_\_\_\_\_ is used.

▶ dot operator (.)

▶ \* operator

▶ à operator

▶ None of given.

Question No: 2 ( M - 1 ) :

What does  $5 \wedge 6$  , evaluate to in decimal where '^' is Exclusive OR operator?

▶ 1

▶ 2

▶ 3

▶ 4

Question No: 3 ( M - 1 ) :

If constructor contains a return statement in its body then compiler will give \_\_\_\_\_

▶ No error

▶ Syntax error

▶ Logical error

▶ Run time error

Question No: 4 ( M - 1 ) :

We can use New keyword inside of Class Constructor.

▶ True

▶ False

Question No: 5 ( M - 1 ) :

When an operator function is define as member function for a Unary operator then the number of argument it take is/are,

▶ Zero

▶ One

▶ Two

▶ N arguments

The declarator of Plus (+) member operator function is

► **Class-Name operator + (Class-Name rhs)**

► operator Class-Name + ()

► operator Class-Name + ( rhs)

► Class-Name operator + ()

Question No: 7 ( M - 1 ) :

Friend function of a class is \_\_\_\_\_ .

► Member function

► **Non-member function**

► Private function

► Public function

Question No: 8 ( M - 1 ) :

We can also create an array of user define data type.

► **True**

► False

Question No: 9 ( M - 1 ) :

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

► Only block of memory is deallocated for objects

► Only destructor is called for objects

► **Memory is deallocated first before calling destructor**

► Destructor is called first before deallocating memory

Question No: 10 ( M - 1 ) :

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

► **True**

► False

Question No: 11 ( M - 1 ) :

Which of the following data type(s) can operate on modulus operator '%'?

► float, int

► float, double

► **int**

► char

Question No: 12 ( M - 1 ) :

Array is passed by value to a function by default.

► True

► False

Question No: 13 ( M - 1 )

With template function, the compiler automatically detects the passed data and generates a new copy of function using passed data.

► True

► False

Question No: 14 ( M - 1 )

What will be the correct syntax to initialize all elements of two-dimensional array to value 0?

► `int arr[2][3] = {0,0} ;`

► `int arr[2][3] = {{0},{0}} ;`

► `int arr[2][3] = {0},{0} ;`

► `int arr[2][3] = {0} ;`

Question No: 15 ( M - 1 )

When an operator function is define as member function then operand on the left side of operator must be an object.

► True

► False

Question No: 16 ( M - 1 )

break statement can be used outside a loop or switch statement.

► True

► False

Question No: 17 ( M - 1 )

The keyword \_\_\_\_\_ is used to return some value from a function.

► return

► break

► continue

► goto

**Question No: 18 ( M - 1 )** :

Every data member or function inside the structure is \_\_\_\_ by default whereas everything declared inside a class is \_\_\_\_ by default.

- ▶ private, public
- ▶ **public, private**
- ▶ private, protected
- ▶ public, protected

**Question No: 19 ( M - 1 )** :

Which of the following is true for the C++ statement given below?

**int &ref = val ;**

- ▶ It creates a synonym for variable 'val'
- ▶ It creates an alias for variable 'val'
- ▶ It's a new name for variable 'val'
- ▶ **All of the given options**

**Question No: 20 ( M - 1 )** :

If overloaded plus operator is implemented as non-member function then which of the following statement will be true for the statement given below?

**obj3 = obj1 + obj2 ;**

- ▶ obj2 will be passed as an argument to + operator whereas obj1 will drive the + operator
- ▶ obj1 will drive the + operator whereas obj2 will be passed as an argument to + operator
- ▶ **Both objects (obj1, obj2) will be passed as arguments to the + operator**
- ▶ Any of the objects (obj1, obj2) can drive the + operator

**Question No: 21 ( M - 1 )** :

Unary operator implemented as member function takes \_\_\_\_ arguments whereas non-member function takes \_\_\_\_ arguments.

▶ One, zero

▶ **Zero, one**

▶ One, two

▶ Two, one

**Question No: 22 ( M - 1 )** :

The input/output streams cin and cout are \_\_\_\_\_ therefore have \_\_\_\_\_.

- ▶ Structures, function
- ▶ Objects, member functions
- ▶ Functions, objects
- ▶ **None of the given options**

**Question No: 23 ( M - 1 )** .

If a symbolic constant has been defined, it will be an error to define it again.

- ▶ **True**
- ▶ False

**Question No: 24 ( M - 1 )** .

Every class contains \_\_\_\_\_.

- ▶ **Constructor**
- ▶ Destructor
- ▶ Both a constructor and a destructor
- ▶ None of the given options

**Question No: 25 ( M - 1 )** .

new and delete keywords are \_\_\_\_\_ in C++ language.

- ▶ Built-in- Function
- ▶ Operators
- ▶ **Memory Allocation Function**
- ▶ None of the given options

**Question No: 26 ( M - 1 )** .

Consider the following code segment.

```
class M {
public:
 M &operator+(const M &);
...
};
```

`p + q` //code of line implies that `p.operator+(q)`

...

Let assume if p and q are class objects then function is implemented as \_\_\_\_\_

► Member function

► Non-member function

► Friend function

► None of the given options

Question No: 27 ( M - 1 )

Assignment operator is -----associative.

► right

► left

► binary

► unary

[Assignment operators](#) in [imperative programming languages](#) are usually defined to be right-associative.

( Referenced by: asad ali ( [asad\\_call2002@yahoo.com](mailto:asad_call2002@yahoo.com) )

Question No: 28 ( M - 1 )

Static variable which is defined in a function is initialized \_\_\_\_\_.

► Only once during its life time

► Every time the function call

► Compile time of the program

► None of the above

[www.vuzs.info](http://www.vuzs.info)

Question No: 29 ( M - 1 )

We can not define a variable of user-defined data type in the class.

► True

► False

Question No: 30 ( M - 1 )

A constructor that will create a new object with a full copy of the other object, is copy is known as \_\_\_\_\_

- ▶ **deep copy**
- ▶ shallow copy
- ▶ constructor copy
- ▶ none of the options

Remember Own Prayer