

Question No: 1 (Marks: 1) - Please choose one
Which one of the following terms must relate to polymorphism?

- ? Static allocation
- ? Static typing
- ? [Dynamic binding](#)
- ? Dynamic allocation

Question No: 2 (Marks: 1) - Please choose one
Multiple inheritance can be of type

- ? Public
- ? Private
- ? Protected
- ? [All of the given](#)

Question No: 3 (Marks: 1) - Please choose one
When a subclass specifies an alternative definition for an attribute or method of its superclass, it is _____ the definition in the superclass.

- ? overload
- ? **overriding**
- ? copy riding
- ? none of given

Question No: 4 (Marks: 1) - Please choose one
Like template functions, a class template may not handle all the types successfully.

- ? **True**
- ? False

Question No: 5 (Marks: 1) - Please choose one
It is sometimes useful to specify a class from which no objects will ever be created.

- ? True
- ? **False**

Question No: 6 (Marks: 1) - Please choose one
Assume a class Derv that is privately derived from class Base. An object of class Derv located in main() can access

- ? public members of Derv.
- ? protected members of Derv.
- ? private members of Derv.
- ? protected members of Base.

Question No: 7 (Marks: 1) - Please choose one
A pointer to a base class can point to objects of a derived class.

- ? True
- ? False

Question No: 8 (Marks: 1) - Please choose one
A copy constructor is invoked when

- ? a function do not returns by value.

- ? an argument is passed by value.
- ? a function returns by reference.
- ? an argument is passed by reference.

- Please choose one

A function call is resolved at run-time in_____

- ? non-virtual member function.
- ? virtual member function.
- ? Both non-virtual member and virtual member function.
- ? None of given

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

When the base class and the derived class have a member function with the same name, you must be more specific which function you want to call (using _____).

- ? scope resolution operator

- ? dot operator
- ? null operator
- ? Operator overloading

Question No: 11 (Marks: 1) - Please choose one
Each try block can have _____ no. of catch blocks.

- ? 1
- ? 2
- ? 3
- ? As many as necessary.

Question No: 12 (Marks: 1) - Please choose one
Two important STL associative containers are _____
and _____.

- ? set,map
- ? sequence, mapping
- ? setmet,multipule
- ? sit,mat

Question No: 13 (Marks: 1) - Please choose one
The mechanism of selecting function at run time
according to the nature of calling object is called,

- ? late binding
- ? static binding
- ? virtual binding
- ? None of the given options

Question No: 14 (Marks: 1) - Please choose one
An abstract class is useful when,

- ? We do not derive any class from it.
- ? There are multiple paths from one derived class to another.
- ? We do not want to instantiate its object.
- ? You want to defer the declaration of the class.

Question No: 15 (Marks: 1) - Please choose one
Which of the following is incorrect line regarding function template?

- ? `template<class T>`
- ? `template <typename U>`
- ? `template < class T, class U>`

Question No: 16 (Marks: 1) - Please choose one
Which of the following is/are advantage[s] of generic programming?

- ? Reusability
- ? Writability
- ? Maintainability
- ? All of given

Question No: 17 (Marks: 1) - Please choose one
By default the vector data items are initialized to _____

- ? 0
- ? 0.0
- ? 1
- ? null

Question No: 18 (Marks: 1) - Please choose one
Which one of the following functions returns the total number of elements in a vector.

- ? length();
- ? size();
- ? ele();
- ? veclen();

Question No: 19 (Marks: 1) - Please choose one
Suppose you create an uninitialized vector as follows:

```
vector<int> evec;
```

After adding the statment,

```
evec.push_back(21);
```

what will happen?

- ? The following statement will add an element to the start (the back) of evec and will initialize it with the value 21.
- ? The following statement will add an element to the center of evec and will reinitialize it with the value 21.
- ? The following statement will delete an element to the end (the back) of evec and will reinitialize it with the value 21.

? The following statement will add an element to the end (the back) of `vec` and initialize it with the value 21.

Question No: 20 (Marks: 1) - Please choose one
An STL container can not be used to,

- ? hold objects of class `employee`.
- ? store elements in a way that makes them quickly accessible.
- ? compile c++ programs.
- ? organize the way objects are stored in memory

Question No: 21 (Marks: 1) - Please choose one
Algorithms can only be implemented using STL containers.

- ? True
- ? False

Question No: 22 (Marks: 1) - Please choose one
The main function of scope resolution operator (`::`) is,

? To define an object

? To define a data member

? To link the definition of an identifier to its declaration

? To make a class private

Question No: 23 (Marks: 1) - Please choose one
When is a constructor called?

? Each time the constructor identifier is used in a program statement

?

new object

? During the construction of a new class

? At the beginning of any program execution

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

Consider the code below,

```
class Fred {
```

```
public:
```

```
Fred();
```

```
...
```

```
};
```

```
int main()
```

```
{  
Fred a[10];  
Fred* p = new Fred[10];  
...  
}
```

Select the best option,

? Fred a[10]; calls the default constructor 09 times

Fred* p = new Fred[10]; calls the default constructor 10 times

? Produce an error

? Fred a[10]; calls the default constructor 11 times

Fred* p = new Fred[10]; calls the default constructor 11 times

? Fred a[10]; calls the default constructor 10 times

Fred* p = new Fred[10]; calls the default constructor 10 times

Question No: 25 (Marks: 1) - Please choose one
Associativity can be changed in operator overloading.

? True

? False

Question No: 26 (Marks: 1) - Please choose one
A normal C++ operator that acts in special ways on newly defined data types is said to be

- ? glorified.
- ? encapsulated.
- ? **classified.**
- ? overloaded.

Question No: 27 (Marks: 1) - Please choose one
Which operator can not be overloaded?

- ? The relation operator (>=)
- ? Assignment operator (=)
- ? Script operator ([])
- ? **Conditional operator (? :)**

Question No: 28 (Marks: 1) - Please choose one
Suppose obj1 and obj2 are two objects of a user defined class A. An + operator is overloaded to add obj1 and obj2 using the function call obj1+obj2.
I identify the correct function prototype against the given call?

- ? A operator + (A &obj);
- ? int + operator();
- ? **int operator (plus) ();**
- ? A operator(A &obj3);

Question No: 29 (Marks: 1) - Please choose one
Default constructor is such constructor which either has no -----or if it has some parameters these have -----
- values

- ? Parameter, temporary
- ? Null, Parameter
- ? **Parameter, default**
- ? non of the given

- Please choose one
Public methods of base class can ----- be accessed in

its derived class

? directly

? indirectly

? simultaneously

? non of the given

Question No: 31 (Marks: 1)

Is Deque a Birectional Container?

Yes, deque behaves like queue (line) such that we can add elements on both sides of it.

Question No: 32 (Marks: 1)

What is meant by Generic Programming?

Generic programming refers to programs containing generic abstractions general code that is same in logic for all data types like printArray function), then we instantiate that generic program abstraction (function, class) for a particular data type, such abstractions can work with many different types of data.

Question No: 33 (Marks: 2)

Sort the following data in the order in which compiler searches a function?

Complete Specialization, Generic Template, Partial Specialization, Ordinary Function.

Specializations of this function template, instantiations with specific types, can be called just like an ordinary function:

```
cout
```

```
<<  
max  
(  
3, 7  
)  
;
```

```
// outputs 7
```

The compiler examines the arguments used to call `max` and determines that this is a call to `max(int, int)`. It then instantiates a version of the function where the parameterizing type `T` is `int`, making the equivalent of the following function:

```
int  
max  
(  
int  
x,  
int  
y  
)
```

```
{  
return  
x  
<  
y  
?  
y  
:  
x  
;  
}
```

the C++ Standard Template Library contains the function `templatemax(x, y)` which creates functions that return either `x` or `y`, whichever is larger. `max()` could be defined like this:

```
template  
  
<  
typename  
T  
>  
T max  
(  
T x, T y  
)
```



```
{  
  
return  
x  
<  
y  
?  
y  
:  
x  
;  
}
```

Question No: 34 (Marks: 2)

State any conflict that may rise due to multiple inheritance?

The conflict may arise is the diamond problem, which our author likes to call the "diamond of doom". This occurs when a class multiply inherits from two classes which each inherit from a single base class. This leads to a diamond shaped inheritance pattern.

For example, consider the following set of classes:

class PoweredDevice

```
{  
};
```

class Scanner: public PoweredDevice

```
{
```

```
};  
classPrinter: publicPoweredDevice  
{  
};  
classCopier: publicScanner, publicPrinter  
{  
};
```

Scanners and printers are both powered devices, so they derived from PoweredDevice. However, a copy machine incorporates the functionality of both Scanners and Printers.

Ambiguity also cause problem.

Question No: 35 (Marks: 3)

Describe three properties necessary for a container to implement Generic Algorithms.

If you declare a container as holding pointers, you are responsible for managing the memory for the objects pointed to. The container classes will not automatically free memory for these objects when an item is erased from the container.

Container classes are expected to implement methods to do the following:

.

create a new empty container (constructor),

.

report the number of objects it stores (size),

.

delete all the objects in the container (clear),

.

insert new objects into the container,

.

remove objects from it,

.

provide access to the stored objects.