## [CS304 Object Oriented Programming Final Paper 2010 unsolved](http://www.vuzs.net/old-papers/328-papers-cs304-object-oriented-programming/1535-cs304-oop-finalterm-spring-2010.html)

CS304 - Object Oriented Programming - Q. No: 1   
Classes like TwoDimensionalShape and ThreeDimensionalShape would normally be concrete, while classes like Sphere and Cube would normally be abstract.  
True  
False

CS304 - Object Oriented Programming - Q. No: 2   
Virtual functions allow you to  
create an array of type pointer-to-base class that can hold pointers to derived classes.  
create functions that can never be accessed.  
group objects of different classes so they can all be accessed by the same function code.  
use the same function call to execute member functions of objects from different classes

CS304 - Object Oriented Programming - Q. No: 3   
A pointer to a base class can point to objects of a derived class.  
True  
False

CS304 - Object Oriented Programming - Q. No: 4   
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.

CS304 - Object Oriented Programming - Q. No: 5   
Each try block can have \_\_\_\_\_\_ no. of catch blocks.  
1  
2  
3  
As many as necessary.

CS304 - Object Oriented Programming - Q. No: 6   
Non Template Friend functions of a class are friends of \_\_\_\_\_\_\_\_instance/s of that class.  
All  
One specific  
All instances of one date type  
None of the given options

CS304 - Object Oriented Programming - Q. No: 7   
Template functions use \_\_\_\_\_\_\_\_\_ than ordinary functions.  
Greater Memory  
Lesser Memory  
Equal Memory  
None of the given options  
  
CS304 - Object Oriented Programming - Q. No: 8   
The find() algorithm  
finds matching sequences of elements in two containers.  
finds a container that matches a specified container.  
takes iterators as its first two arguments.  
takes container elements as its first two arguments.

CS304 - Object Oriented Programming - Q. No: 9  www.vuzs.net  
The copy() algorithm returns an iterator to  
the last element copied from.  
the last element copied to.  
the element one past the last element copied from.  
the element one past the last element copied to.

CS304 - Object Oriented Programming - Q. No: 10   
If you define a vector v with the default constructor, and define another vector w with a one-argument constructor to a size of 11, and insert 3 elements into each of these vectors with push\_back(), then the size() member function will return \_\_\_\_\_\_ for v and \_\_\_\_\_ for w.  
11 for v and 3 for w.  
0 for v and 0 for w.  
0 for v and 3 for w.  
3 for v and 11 for w.

CS304 - Object Oriented Programming - Q. No: 11   
Which is not the Advantage of inheritance?  
providing class growth through natural selection.  
facilitating class libraries.  
avoiding the rewriting of code.  
providing a useful conceptual framework.

CS304 - Object Oriented Programming - Q. No: 12   
class DocElement  
{  
public:  
virtual void Print() { cout <  
<  
};  
class Heading : public DocElement  
{  
public:  
void Print() { cout <  
<  
};  
class Paragraph : public DocElement  
{  
public:  
void Print() { cout <  
<  
};  
void main()  
{  
DocElement \* p = new Paragraph();

p->Print();  
}

When you run this program, it will print out a single line to the console output.  
What will be in that line?  
Select one correct answer from the following list:

Generic element  
Heading element  
Paragraph element  
Nothing will be printed.

CS304 - Object Oriented Programming - Q. No: 13   
Which type of inheritance is being represented by the following statement,  
class X : public A, public B { ... ... };  
Single inheritance  
Multiple inheritance  
Double inheritance  
None of the given options

CS304 - Object Oriented Programming - Q. No: 14   
When we write a class template the first line must be:  
template  
template  
template  
Here T can be replaced with any name but it is preferable.  
class class-name()  
class template  
<

CS304 - Object Oriented Programming - Q. No: 15   
Function templates should be used where code and behavior must be identical.   
True  
False  
Rational

This will help us to standardized the codes.  
They are meant for the situations where same functionality can be adapted to more than one type or class. It will reduce the need of repeating the entire code for each type.  
(zubair, vuzs.net,Jul2011)

CS304 - Object Oriented Programming - Q. No: 16   
Which of the following is/are advantage[s] of generic programming?  
Reusability  
Writability  
Maintainability  
All of given

CS304 - Object Oriented Programming - Q. No: 17   
The specialization pattern after the name says that this specialization is to be used for every,  
data type  
meta type  
virtual type  
pointer type

CS304 - Object Oriented Programming - Q. No: 18   
A range is often supplied to an algorithm by two \_\_\_\_\_\_\_ values.  
italic  
iteration  
iterator  
None of given

CS304 - Object Oriented Programming - Q. No: 19   
Which of the following is an integral part of an object?  
State  
Behavior  
Unique identity  
All of the given

CS304 - Object Oriented Programming - Q. No: 20   
Consider the following statement  
Cupboard has books  
What is the relationship between Cupboard and books?  
Composition  
Aggregation  
Inheritance  
None of the given options

CS304 - Object Oriented Programming - Q. No: 21   
Which sentence clearly defines an object?  
one instance of a class.  
another word for a class.  
a class with static methods.  
a method that accesses class attributes.

CS304 - Object Oriented Programming - Q. No: 22   
\_\_\_\_\_\_\_\_\_\_\_, which means if A declares B as its friend it does NOT mean that A can access private data of B. It only means that B can access all data of A.  
Friendship is one way only  
Friendship is two way only  
NO Friendship between classes  
Any kind of friendship

CS304 - Object Oriented Programming - Q. No: 23   
The statement objA=objB; will cause a compiler error if the objects are of different classes.  
True  
False

CS304 - Object Oriented Programming - Q. No: 24   
Consider the call given below of an overloaded operator "+",  
Rational\_number\_1 + Rational\_number\_2  
Where Rational\_number\_1 and Rational\_number\_2 are the two objects of Rational\_number class (a user defined class). Identify which of the above two objects will be passed as an argument to the overloaded operator function?  
Rational\_number\_1  
Rational\_number\_2  
Both Rational\_number\_1 & Rational\_number\_2  
any of the two objects, randomly

CS304 - Object Oriented Programming - Q. No: 25   
If a class D has been derived using protected inheritance from class B (If B is a protected base and D is derived class) then public and protected members of B -------- accessed by member functions and friends of class D and classes derived from D  
can be  
cannot be  
does restirct to be  
not given

CS304 - Object Oriented Programming - Q. No: 26   
In Private -------------- only member functions and friend classes or functions of a derived class can convert pointer or reference of derived object to that of parent object  
specialization  
inheritance  
abstraction  
composition

CS304 - Object Oriented Programming - Q. No: 27 ( m a r k s: 2 )  
Give two uses of a destructor.

CS304 - Object Oriented Programming - Q. No: 28 ( m a r k s: 2 )  
Descibe the way to declare a template class as a friend class of any other class.

CS304 - Object Oriented Programming - Q. No: 29 ( m a r k s: 2 )  
Give the name of two basic types of containers collectively called First class containers?

CS304 - Object Oriented Programming - Q. No: 30 ( m a r k s: 2 )  
State any conflict that may rise due to multiple inheritance?

CS304 - Object Oriented Programming - Q. No: 31 ( m a r k s: 3 )  
What will be the output after executing the following code?

class c1{  
public:  
virtual void function(){  
cout<  
<  
}

};  
class c2: public c1{  
public:  
void function(){  
cout<  
<

}

};  
class c3: public c1 {  
public:  
void function(){  
cout<  
<  
}

};

int main(){

c1 \* test1 = new c2();  
c1 \* test2 = new c3();  
test1->function();  
test2->function();  
system(“PAUSE”);  
return 0;  
}

CS304 - Object Oriented Programming - Q. No: 32 ( m a r k s: 3 )  
If we declare a function as friend of a template class will it be a friend for a particular data type or for all data types of that class.

CS304 - Object Oriented Programming - Q. No: 33 ( m a r k s: 3 )  
Tell the logical error/s in the code given below with reference to resource management; also describe how we can correct that error/s.

class Test{

public:  
int function1(){  
try{  
FILE \*fileptr = fopen(“filename.txt”,“w”);  
throw exception();  
fclose(fileptr);  
return 0;  
}  
catch(Exception e){  
...  
}  
}  
};

CS304 - Object Oriented Programming - Q. No: 34 ( m a r k s: 5 )  
What is the output produced by the following program?

#include

void sample\_function(double test) throw (int);

int main()  
{  
try  
{  
cout <  
<  
sample\_function(98.6);  
cout <  
<  
}  
catch(int)  
{  
cout <  
<  
}

cout <  
<  
return 0;  
}  
void sample\_function(double test) throw (int)  
{  
cout <  
<  
if(test < 100)  
throw 42;  
}

CS304 - Object Oriented Programming - Q. No: 35 ( m a r k s: 5 )  
The code given below has one template function as a friend of a template class,  
1. You have to identify any error/s in this code and describe the reason for error/s.  
2. Give the correct code after removing the error/s.

template  
void Test(U);  
template

class B {  
int data;  
public:  
friend void Test( T );  
};

template  
void Test(U u){  
B b1;  
b1.data = 7;  
}  
int main(int argc, char \*argv[])  
{  
char i;  
Test(i);  
system("PAUSE");   
return 0;  
}

CS304 - Object Oriented Programming - Q. No: 36 ( m a r k s: 5 )  
Consider the following class,  
class Base  
{  
char \* p;  
public:  
Base() { p = new char[10]; }

~Base() { delete [] p; }  
};  
class Derived : public Base  
{  
char \* q;  
public:  
Derived() { q = new char[20]; }

~Derived() { delete [] q; }  
};  
void foo()  
{  
Base\* p = new Derived();

delete p;  
}

With this program, every time function foo is called, some memory will leak.  
Explain why memory will leak. Also, explain how to fix this problem.