PAPER CS304
DATED 18-07-2012

question # 1
A programmer implements virtual functions by the use of virtual keyword, like:

```
virtual void display(){
.................
}
```

but how the virtual functions are implemented by compiler?

===

Question # 2
Consider the following complex class,

```
class Complex{
private:
      double real, img;
public:
    ........
       };
```

we want to overload subtraction(-) operator for our complex class such that we can subtract any double value from our complex class object using any one of the statements given below,

```
Complex c1;
complex c2 = 1.546-c1;
complex c3 = c1-2.456;
```

you have to give the c++ code of overloaded subtraction (-) operator to perform thesekind of operations.

==============================

Question # 3
What are Iterators? explain input iterator and output iterator.
Ans:iterators are internal data member of container. That can traverse

a container without exposing its internal representation. Iterators are for containers exactly like pointers are for ordinary data structures.

Input Iterators:

Using input iterators we can read an element, and we can only move in forward direction one element at a time, these can be used in implementing those algorithms that can be solved in one pass e.gmoving container once in single direction from start to end like find algorithm.

Output Iterators :

Using output iterators we can read an element, and we can only move in forward

direction one element at a time, these can be used in implementing those algorithms that can be solved in one pass e.gmoving container once in single direction from start to end like find algorithm .

==================================

Question # 4
Consider the code below,

```
1. Template<typename T> class Test {
2. T value;
3. public:
4. Test( T val) : value(val) {}
5 Test (Const Test<T>& C): value (c.value) {}
6. }:
7. int main() {
8. Test<int>t1(0), t2(0);
9. Test<float> t3 =t1;
10. system("pause");
11. Return 0;
12. }
```

Given line number having any error/errors in this code ecplain the reason fro error/s and give corrected code as well

Next quesiton

What ll be output after excuting following line of code, (suppose there is no error and code will be executed correctely, justify your answer as well,

```
class student{
      staticintnoOf students;
     introllNo;
public:
      student::Student(){
```

```
                noOfStudents++;
}
Student::~Student(){
        NoOfStudents--;
}
     staticintgetTotalStudent(){
               returnnoOfStudents;
}
};
int Student::noOfStudents=0;
int main(){
 Student obj1, obj2;
 {
      student obj3;
}
cout<<student::getTotalstudent();
}
```

Next question,

Write c++ code that uses try catch block in member initialization list of any class.

Ans:

```
Student::Student (String aName)
try : name(aName) {
...
}
catch(…) {
}
```

next question

Give the difference between virtual inheritance and multiple inheritance.

Ans:

Multiple Inheritance is bassically a process , The process when a child can be derived from more than one parent class.And If more than one base class have a function with same signature then the child will have two copies of that function; Calling such function will result in ambiguity.

Example of multiple inheritance:

Suppose we have a changeGearmethod in Vehicle class that is applicable to bothwater and
land vehicle, we also have Float and Move methods in water and land vehicles respectively
then our amphibious vehicle will have all these methods,

While in In virtual inheritance there is exactly one copy of the anonymous base class object

Example of virtual inheritance:

```
class Vehicle{
protected:
int weight;
};
classLandVehicle :
public virtual Vehicle{
};
classWaterVehicle :
public virtual Vehicle{
};
Example
classAmphibiousVehicle:
publicLandVehicle,
publicWaterVehicle{
public:
AmphibiousVehicle(){
weight = 10;
}



};
```

======================================

next question

Differentiate between specialization and sub-typing by giving most important reson/s

Ans:

Specialization means that derived class is behaviorally incompatible with the base class Behaviorally incompatibility means that base classcan't always be replaced by the derived class Derived class has some different of restricted characteristics than of base class.

Example:Suppose we want to add one more class of Adult for some special requirement like for ID card generation such that it is a person butits age is greater than 18 and having all other behavior of that of person class.in this case the best solution is that we derive adult class from person class and restrict age in that class.

Sub-typing means that derived class is behaviorally compatible with the base class Derived class has all the characteristics of base class plus some extra characteristicsBehaviorally compatible means that base class can be replaced by the derived class.

Example:

Student has two extra attributes program and studyYearSimilarly it has extended behaviour by adding study and takeExam.

==============================
next question explain the statment below

vector<int>ivec(4.3);

Ans:
in this case Vector instance will store our data .integers 4.3

next question
give at least one advantage and on disadvantage of templates.

Advantage:
can drastically reduce development time using templates in combination with STL. On a more fundamental level, templates allow you to write a single algorithm and use it across different data types.

Disadvantage:templates can't be linked and distributed as a library. They must be compiled at compile time, and, therefore, all implementations of a template algorithm must be included in the header files in their entirety.

nextqustion
what do you know about function templates? describe in berefly.

Ans:

We can use Function Templates where we want to write general function like
printArray. It **can be parameterized to operation different types of data types.**

<u>Declaration:</u>

template< class T >
voidfunName(T x);

<u>Example:</u>

template<typename T>
voidprintArray(T*array, int size)
{

```
for ( int i = 0; i < size; i++ )
cout<< array[ i ] << ", "; // here data type of array is T
}
```

next question

find out adn correct the syntax error(s) in the given code snippet?

```
template<classname T>
classTemplate_class{
private:
T data;
//...
Public:
///...
Void input ();
};
Template<class T>
voidTemplate_class::input(){
cin>>data;
}void main()
{
Template_class<int>obj;
obj.input();
}
```