**CS304 Quiz No 3 with solution**

A template provides a convenient way to make a family of

Select correct option:
variables and data members
**functions and classes**
classes and exceptions
programs and algorithms

A class template may inherit from another class template.
Select correct option:
**True**
False

Target of a *\_\_\_\_\_* function call is determined at run time.
Select correct option:
instance
**virtual**
operator
none of given

A class hierarchy
Select correct option:
shows the same relationships as an organization chart.
describes “has a” relationships.
**describes “is a kind of” relationships.**
shows the same relationships as a family tree.

Sender of the message does not need to know the exact class of receiver in\_*\_\_\_\_\_*.
Select correct option:
Abstraction
Polymorphism
Inheritance
none of the given

A function call is resolved at run-time in\_*\_\_\_\_\_\_\_\_*
Select correct option:
non-virtual member function
virtual member function
**Both non-virtual member and virtual member function.**
None of given

Adding a derived class to a base class requires fundamental changes to the base class.
Select correct option:
True
**False**

User can make virtual table explicitly.
Select correct option:
**True**
False

Binding means that target function for a call is selected at compile time.
Select correct option:
**Static**Dynamic
Automatic
None of given

Target of a *\_\_\_\_\_* function call is determined at run time.
Select correct option:
instance
**virtual**
operator
none of given

Which line will produce error. Class phone: private Transmit, private Receiver { } 1. int main() 2. { 3. phone obj; 4. Tranmit\* obj1 = &obj; 5. Received obj2 = &obj; 6. }
Select correct option:
3rd line will produce error
4th line will produce error
3rd and 4th line will produce error.
5th line will produce error

Function overriding is done in context of,
Select correct option:
Single class
Single derived class
Single base class
**Derived and base classes**

Consider the code below, class class1{ public: void func1(); }; class class2 : protected class1 { }; Function func1 of class1 is *\_\_\_\_* in class2,
Select correct option:
public
**protected**
private
none of the given options

the following statements: 1) int iArray[5]; 2) int \*pArr = iArray;
Select correct option:
These statements will compile successfully
Error in first statement
**Error in second statement**
None of given options

Methodologies to the development of reusable software relate to\_*\_\_\_\_\_\_\_*.
Select correct option:
Structure programming
procedural programming
**generic programming**None of the given

function template must have a parameter.
Select correct option:
True
False

The default inheritance mode is,
Select correct option:
Public inheritance
Protected inheritance
**Private inheritance**None of these options

Two functions with same names, parameters and return type can exist in,
Select correct option:
**Function overloading**Function overriding
Operator overloading
None of these options

Consider the code below, class c1{ }; class c2 : public c1 { }; class c3 : public c2 { }; Then c2 is,
Select correct option:

**Direct base class of c3**
Direct child class of c3
Direct base class of c1
None of these

Virtual functions allow you to
Select correct option:
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.

User can make virtual table explicitly.
Select correct option:
True
**False**

In order to define a class template the first line of definition must be:
Select correct option:
**template <typename T>**
typename <template T>
Template Class <ClassName>
Class <Template T>

Consider the following statements: 1) int iArray[5]; 2) int \*pArr = iArray;
Select correct option:
These statements will compile successfully
Error in first statement
**Error in second statement**None of given options

In c++ dynamic binding and polymorphism will be achieved when member function will be *\_\_*.
Select correct option:
private
public
**virtual**
inline

In type in depended function template should be use where code and behavior must be identical.
Select correct option:
**True**False