COMSATS Institute of Information Technology (CIIT)
Islamabad Campus
Object Oriented Programming
BCS-2 Fall 2009

Instructor: Dr. Zia-ud-Din Total Marks: 100 Allocated Time: 3 hours

Name: _____ Registration Number: _____

Attempt all questions.

Q. 1: Complete the following statements. (20*1=20)

1. Inheritance is called a _____ relationship.
2. A _____ function of a class can be called without instantiating any object.
3. _____ , _____ and _____ are three ways to achieve polymorphism in C++.
4. Friend functions are against _____ principle of object-oriented programming.
5. Pointers of _____ class can point to objects of_____ as well as _____ class.
6. Aggregation is called a _____ relationship.
7. The compiler gives an error when we try to instantiate an object of _____ class.
8. _____ , _____ and _____ are three C++ keywords associated with exceptions.
9. Base class destructors should always be _____.
10. Virtual base classes are related to _____.
11. _____ functions can make a bridge between two or more different classes.
12. An abstract class contains at least one _____ function.
13. The function to overload division operator (/) for class X takes _____ argument(s).
14. _____ loop is used when you want a guarantee that the loop body is executed at least once.
15. A static function _____ access non-static data members.
16. The following statement invokes _____ for class MyClass (Assume m1 is already declared).

    MyClass m2 = m1;
17. _____ , _____ and _____ are three pillars of object-oriented programming.
18. The only way to initialize constant data members of a class is through _____.
19. _____ is a stronger form of aggregation.
20. _____ is one of the operators which cannot be overloaded.

Q. 2: Show the output or find the errors in each of the following cases. Assume all needed header files and namespaces are there. Also ignore potential errors caused by line breaks or text casing. (5*4=20)

1.
```
void main()
{
    int x=5;
    int y=6;
    cout<<++x+2+y--;
}
```

2.
```
class A
{
public:
    A(){cout<<"A's constructor called\n";}
    ~A(){cout<<"A's destructor called\n";}
};
class B:public A
{
public:
    B(){cout<<"B's constructor called\n";}
    ~B(){cout<<"B's destructor called\n";}
};
void main()
{
    A a;
    B b;
}
```

3.
```
class A
{
    int data;
public:
    virtual void Fxn1()=0;
    void Fxn2(){cout<<"I am Fxn2"<<endl;}
};
class B:public A
{
public:
    void Fxn1(){cout<<"I am Fxn1"<<endl;}
};
void main()
{
```

```
    A a;
    B b;
    a.Fxn2();
    b.Fxn1();
}
```

4.
```
class A
{
    int data;
public
    A(){data=0;}
    static void Fxn(){cout<<data<<endl;};
};
void main()
{
    A a;
    A::Fxn();
}
```

5.
```
class B;
class A
{
    int data;
    friend int Add(A,B);
};
class B
{
    int data;
public:
    friend int Add(A,B);
};
int Add(A a,B b){return (a.data+b.data;)
void main()
{
    A a;
    B b;
    cout<<Add(a,b);
}
```

Q. 3: Write brief notes on any 5 of the following with appropriate code examples wherever necessary. (5*6=30)

1. Exception handling in c++
2. Virtual destructors
3. The this pointer
4. Static functions
5. public, private and protected access modifiers
6. Function overloading

Q. 4: Write C++ programs with the following specifications. (10+8+12=30)

1. Write a class Distance with data members feet and inches. Overload assignment operator and write copy constructor for this class. (2+4+4=10)
2. Write a class Movie having data members name, duration and basedOn (basedOn contains the name of the novel if the movie is based on some novel, otherwise it contains an empty string). Write another class named Novel having data members name and author. Write a function which is a friend of both these classes. The friend function takes two objects: one for each class and returns true if the passed movie is based on the passed novel, otherwise it returns false. (2+2+4=8)
3. Write a class A having a single data member AData with a constructor, destructor and a function ShowData (that displays AData). Inherit two classes B and C from A having single data members BData and CData, respectively. Override ShowData in both these classes. In your main program, create an array of A's pointers. Fill this array with different objects of A,B and C. Call ShowData for each element of this array using a for loop. Display the output of your program. (4+3+5=12)

********** Good Luck **********