Object-Oriented Programming (OOP) Sample Questions

Student ID: _____

Name: _____

1. What are the four basic concepts of OOP? (20%)

A_____

E_____

I_____

P_____

2. What are the three main categories of design patterns? (15%)

C_____ patterns

S_____ patterns

B_____ patterns

3. Multiple Choice Questions (40%)

( ) Which one is a creational design pattern?

a. Factory b. Adaptor c. Façade d. Proxy

( ) For a C++ class MyClass{ }, which one is its destructor?

a. MyClass() b. ~MyClass() c. &MyClass() d. *MyClass()

( ) In C++, how to access the parent class's pointer?

a. super b. parent c. base d. there is no keyword to access the parent class's pointer in C++

( ) In Java, how to access the parent class's pointer?

a. super b. parent c. base d. there is no keyword to access the parent class's pointer in Java

( ) In Java, which keyword is used to inherit a class?

a. extends b. extend c. : d. inherit

( ) In C++, which keyword is used to inherit a class?

a. extends b. extend c. : d. inherit

( ) Which keyword is **NOT** used to define the accessibility of class members in C++?

a. public b. private c. protected d. primitive

( ) What is the default class member accessibility?

a. public b. private c. friend d. protected

4. Programming (25%)

What is the output of the following program:

_____

_____

```
#include <iostream>
using namespace std;
// Base class
class Shape {
public:
     Shape(double a = 0, double b = 0) {
          width = a; height = b;
     }
     ~Shape() {}
     virtual double area() = 0;

     double width;
     double height;
};

class Rectangle : public Shape {
public:
     Rectangle(double a = 0, double b = 0) :Shape(a, b) { }
     double area() {
          cout << "Area of Rectangle" << endl;
          return (width * height);
     }
```

```
};

class Triangle : public Shape {
public:
     Triangle(double a = 0, double b = 0) :Shape(a, b) { }

     double area() {
          cout << "Area of Triangle" << endl;
          return (width * height / 2);
     }
};

class Circle : public Shape {
public:
     Circle(double a = 0, double b = 0) :Shape(a, b) { }

     double area() {
          cout << "Area of Circle" << endl;
          return (width * width * 3.1415926);
     }
};

int main(int argc, char *argv[])
{
     double num1 = 5, num2 = 3;
     Shape *shape;
     Rectangle rect;
     Circle circle;
     Triangle triangle;
     shape = &rect;
     shape->width = num1;
     shape->height = num2;

     // Print the area of the object.
     cout << shape->area() << endl;

     return 0;
}
```