**SWENG 421 Software Architecture Lab 1**

Note: a team of up to two persons can work on the lab together.

1. Draw a UML class diagram based on the following descriptions. Please indicate different packages and model the relationships: generalization/realization, association, aggregation, composition, and dependency. Submit the diagram to the Canvas dropbox. (5 pts)

* In package p1, there are two interfaces I1 and I2. I2 is a subclass of I1. In I1, there is a method “void m1()”. In I2, there is a method “int m2(double d)”.
* In package p2, there is an abstract class AC, which is a subclass of I2. It requires a public instance variable c3 of class C3 and a public instance variable c4 of class C4, which is in packge p4. Regarding methods, it has a public abstract method “int m3()”, an implemented private method “int m4()” and an implemented public method “double m5(String[] sa)”.
* In package p3, class C1 and C2 are subclasses of AC. Class C1 has two attributes “private int i” and “public double j”. Class C2 overrides AC’s m5 method.
* In package p4, class C3 requires a package instance variable of C5 to perform computations in two of its methods, i.e., private method “int m6()” and package method “void m6(int val). Besides, it also has a protected method “boolean m7(C4 c4)”.
* In the same package p4, class C5 has a package method “void m8(C6 c6)” and class C6 has a public method “void m9()”. Class C6 consists of two classes C7 and C8 also in package p4. The object of C7 is no longer accessible if the object of C6 becomes null. However, object of C8 may still be accessed if the object of C6 becomes null. Class C7 has a public method “Integer m10(int k)”, while C8 has a public method “int m11(Double d)”.

1. Develop a C# drawing program as shown in the next page. The following lists the requirements.

* Apply polymorphism and dynamic binding for development of this drawing program.
* Provide three buttons for the user to choose or change a desired shape for drawing.
* A shape can be a line, a rectangle or an ellipse, and can be drawn in an arbitrary direction.
* A color based on RGB values can be set to color the subsequent drawn shapes.
* The dynamic binding will call the shape’s drawColoredShape method to draw its own shape based on the set color.
* The system starts with a small window and the user can resize the window without erasing the previously drawn shapes.
* When drawing a shape, the mouse movement will change the size of the drawn shape.

Hints: The Graphics, Bitmap, Color, Pen and Brush classes are likely to be used.

C# Keywords: virtual, override, abstract and internal may be useful. Feel free to check the C# 6.0 Specification and the Java-Csharp\_Comparison.html file posted below Lab 1 on Canvas.

The Bitmap class is good for double buffering technique by using two images as the buffers. The background Bitmap image should be filled up with a color first to prevent undesirable transparent effect.

Bitmap bm = new Bitmap(sizeX, sizeY);

Graphics g = Graphics.FromImage(bm);

Rubrics:

* Due in one week.
  + UML class diagram for Q1. (5 pts)
  + Polymorphism and dynamic binding: (2 pts)
  + Program correctness: (3 pts)
  + Double buffering technique applied: (2 bonus points)
* Compress the entire working folder with the class diagram to a zip file and submit it to Canvas.

