**Activity on Abstract Classes and Polymorphism**

Instructor: Asma Sanam Larik

Q1: Suppose we have an abstract class Shape.java. It contains a single point denoted by integer variable x1, y1 respectively. It has a two argument constructor that initializes the point and a toString() method to display their values. It also contains a pure virtual function computeSurfaceArea(). Next assume a class Triangle that extends from the basic class shape with instance data comprising of three points denoted by x1,y1,x2,y2,x3 and y3 respectively. The class also has a constructor that receives the 3 points and initializes them, a toString method that outputs the 3 points, a computeEachSide() method and a computeSurfaceArea() methods. The computeEachSide() method takes two points and computes the distance between them using Euclidean distance formula. These distances should then be saved as attributes A,B,C respectively. Next for computeSurfaceArea() use the following formulas:

S= (A+B+C) /2

SurfaceArea = sqrt( S (S-A)\* (S-B)\* (S-C))

We want to extend this class to represent a 3-dimensional Pyramid, which consists of 4 Points. Since the Pyramid will consist of in essence, 4 triangles, computeSurfaceArea() for the Pyramid will compute 4 \* the surface area of the Triangle made up of 3 of those 4 Points. Write the Pyramid class to handle the difference(s) between a Pyramid and the previously defined

Triangle. Assume the instance data of Triangle are protected and all methods are public.

------------------------------------------The End ☺-----------------------------------------------