Question 1 [20 points]

Does each of the following programs compile and run? If no, explain the reason; otherwise, give the output.

1. class P1 {
   public static void main(String[] args) {
      B b = new B();
      A a = b;
      a.show();
      b.show();
   }
}

class A {
   public String str = "A";
   public void show() {
      System.out.println(str);
   }
}

class B extends A {
   public String str = "B";
   public void show() {
      System.out.println(str);
   }
}

2. class P2 {
   public static void main(String[] args) {
      B b = new B();
      b.m();
   }
}

class A {
   private int x;
}

class B extends A {
   private int y;

   public void m() {
      super.x = y;
   }
}
3. class P3 {
    public static void main(String[] args){
        A b = new B();
    }
}
class A {
    A(){
        System.out.println("A");
    }
}
class B extends A {
    B(){
        System.out.println("B");
    }
}

4. class P4 {
    public static void main(String[] args) {
        try {
            int value = 30;
            if (value < 40)
                throw new Exception("value is too small");
        }
        catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
        System.out.println("Continue after the catch block");
    }
}

5. class P5 {
    public static void main(String[] args) {
        for (int i = 0; i < 2; i++) {
            System.out.print(i + " ");
            try {
                System.out.println(1 / 0);
            }
            catch (Exception ex) {
            }
        }
    }
}
Question 2: [20 points]

Write a class named Hexagon that extends GeometricObject and has the following specification. Assume all six sides of the hexagon are of equal size.

```java
public class Hexagon extends GeometricObject {
    private double side;

    /** Construct a Hexagon with the specified side */
    public Hexagon(double side) {
        // Implement it
    }

    /** Implement the abstract method findArea in GeometricObject */
    public double findArea() {
        // Implement it ( )
    }

    /** Implement the abstract method findPerimeter in GeometricObject */
    public double findPerimeter() {
        // Implement it
    }
}
```
Question 3: [20 points]

Implement a class named MyArrayList that extends the java.util.ArrayList by overriding the add method in the following way: it does nothing if the object to be added already exists in the collection; otherwise it calls the add method in the super class to add the object into the collection. Use o1.equals(o2) to test if two objects o1 and o2 are equal.
Question 4: [20 points]

Implement the following functions (static methods):

1. public static ArrayList<Object> lastThree(ArrayList<Object> lst): This function returns a list that contains the last three elements of lst. It throws a RuntimeException if lst has fewer than three elements. For example, if lst is ['a','b','c','d'], then it returns ['b','c','d'].

2. public static Object mostFrequent(ArrayList<Object> lst): This function returns the most frequently occurring element in lst. For example, if lst is [1,3,1,3,2,1], then the returned value is 1. If there are more than multiple most-frequent elements, then the function can return any one of them.
Question 5: [20 points]

Suppose that a text file contains an unspecified number of integers separated by blank spaces. Write a program that prompts the user to enter the file name, reads the integers from the file, and displays their total and average.