Question 1 (25 points)

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

1. public class P1 {
   public static void main(String[] args) {
      f(6);
   }
   public static void f(int length) {
      if (length > 1) {
         System.out.print((length - 1) + " ");
         f(length - 1);
      }
   }
}

2. class P2 {
   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) {
            System.out.println("divided by zero");
         }
      }
   }
}
3. import java.util.*;
class P3 {
    ArrayList<Integer> lst = new ArrayList<Integer>();

    public void add(Integer obj){
        int n = lst.size();
        if (n != 0 && lst.get(n-1).equals(obj))
        {
        } else
        lst.add(obj);
    }

    public static void main(String[] args){
        P3 d = new P3();
        d.add(1); d.add(1); d.add(1); d.add(2); d.add(3);
        System.out.println(d.lst);
    }
}

4. public class P4 {
    public static Object max(Object o1, Object o2) {
        if (((Comparable)o1).compareTo(o2) >= 0) {
            return o1;
        } else {
            return o2;
        }
    }

    public static void main(String[] args){
        System.out.println(max(1,2));
    }
}
5. class A {
    public A() {
        System.out.println("A’ constructor");
    } 
    
    public void m() {
        System.out.println("A’s m");
    }
}

public class P5 extends A {
    public P5() {
        System.out.println("P5’s constructor");
    }
    
    public void m() {
        System.out.println("P5’s m");
    }

    public static void main(String[] args) {
        A o = new A();
        ((P5)o).m();
    }
}
Question 2 (25 points)

1. Design a class named Candidate for holding a candidate in an election. Assume that only two attributes of a candidate, namely, the name and the number of votes, are of interest here. The class contains a constructor that initializes the member variables to the given values, and a get method for each of the member variables. The class implements the Comparable interface, and compares two candidates based on their numbers of votes.

2. Designing a class named VotingMachine that contains a collection of candidates and the following methods:

   • addCandidate(String name): This method adds a candidate of a given name to the collection of candidates. This method throws an exception if a candidate of the given name already exists in the collection.

   • castVote(String name): This method casts a vote to the candidate of the given the name. This method throws an exception if there is no candidate of the name in the collection.

   • winner(): This method returns the name of a candidate who received the maximum votes. If there are two or more candidates who received the maximum votes, this method returns any of them.
Question 3 (15 points)

Implement a class named MyArrayList that extends class the java.util.ArrayList. The class MyArrayList overrides the add(obj) method in the following way: it adds obj into the list if the object does not occur in the list or it occurs in the list only once; otherwise, if the object occurs in the list more than once, then the method does nothing.
Question 4 (15 points)

Suppose that a text file, named "src.txt", contains an unspecified number of integers, each occupying a line. Write a program that copies the integers into another file, named "dest.txt", with each integer increased by 1. For example, if "src.txt" has the following lines

1  
23  
456

then the created file "dest.txt" should have the following content:

2  
24  
457

If a file named "dest.txt" already exists, the program throws an exception.
Question 5 (20 points)

Write the following functions (static methods) using recursion:

1. public static String dec2bin(int value): This function takes a non-negative integer and returns its binary representation as a string. For example, the function call dec2bin(6) returns "110".

2. public static boolean isPalindrome(String s): This function returns true if and only if s is a palindrome, meaning that s reads the same backward and forward. For example, "mom" is a palindrome.

3. public static ArrayList<String> gen(int n) (extra 5 points): This function returns a list of all possible strings of a's and b's that contain more a's than b's. For example, for n = 3, it returns ["aaa","aab","aba","baa"]. The order of the strings in the list is not important.