CISC 3115
Final Exam

This exam consists of 5 questions. Please complete the exam and submit it as a plain text email with the subject "CISC 3115 Final Exam" to nzhou@brooklyn.cuny.edu by 7PM on Monday, May 24.

Question 1:

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) {
      System.out.println(f(6, ""));
   }
   public static String f(int n, String acc) {
      return (n >= 1) ? f(n - 1, n+acc) : acc;
   }
}

2. class P2 {
   public static void main(String[] args) {
      int s = 0;
      for (int i = 0; i < 6; i++) {
         s += i;
         try {
            System.out.println(1 / 0);
         }
         catch (Exception ex) {
         }
      }
      System.out.println(s);
   }
}
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(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 P5();
        ((P5)o).m();
    }
}
Question 2

Implement the class, Time, specified below:

class Time {
    private int hour;
    private int minute;

    public Time();
    public Time(int hour, int minute);

    public int getHour();
    public int getMinute();

    public void set(int hour, int minute);
    public void add(int minutes);

    public String toString();
    public String toString12();
}

A Time object represents a 24-hour clock time, in which hour is in the range 0..23, and minute is in the range 0..59.

- The no-arg constructor initializes both hour and minute to 0, and the other constructor initializes hour and minute to the specified values.
- The get and set methods are for retrieving and updating the values of the member variables.
- The method add adds the specified number of minutes to the time.
- The toString converts the time object to a string of the format hour:minute.
- The toString12 converts the time object to a string of the format hour:minute AM or hour:minute PM.
Question 3

Implement a class named MyArrayList that extends class the java.util.ArrayList. The class MyArrayList has the following two differences from ArrayList:

- 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.

- MyArrayList overrides the toString method such that parentheses () are used, rather than brackets []. For example, for a list of three elements 1, 2, and 3, the returned string is "(1, 2, 3)", not "[1, 2, 3]".
Question 4

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

Write the following functions (static methods) using recursion:

1. public static int sumDigits(int value): This function takes a non-negative integer and returns the sum of its digits. For example, for value = 12345, the output is 15.

2. public static String prettyString(int value): This function takes a non-negative integer and returns the digits of the integer as a string of the format that separates groups of three (counting from the right) with commas. For example, when value = 12345, the output should be "12,345".

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 fewer a’s than b’s. For example, for n = 3, it returns ["bbb","bba","bab","abb"]: The ordering of the strings in the list is not important.