

# ECE122 Introduction to ECE II

**Spring 2008**  
**2<sup>nd</sup> Midterm Examination**  
**(120 minutes, closed book)**

NAME: \_\_\_\_\_ SOLUTION SET \_\_\_\_\_

Student ID: \_\_\_\_\_

All answers must be written in Java using techniques learned from lectures 1-15 in the course

| <b>Question</b>            | <b>Score</b> |
|----------------------------|--------------|
| 1 (20 points )             |              |
| 2 (15 points)              |              |
| 3 (20 points)              |              |
| 4 (15 points)              |              |
| 5 (15 points)              |              |
| 6 (15 points)              |              |
| <b>Total (100 points )</b> |              |

## 1. (20 points)

Write a class called **RandomDouble** with the following two methods:

One method, called **randomPower**, takes a one dimensional array of integers **param\_int** as a parameter and returns a one dimensional array of integers. The method creates a second array, **random\_int**, of the same size and fills it with random positive integer values. The method then raises each integer in the **param\_int** array to the power of the random integer with the same index (e.g. raise **param\_int** [0] to the power of **random\_int**[0]) and stores these values in a third array, **return\_array**. This array should be returned by the method. Your method must work on arrays of any size.

Write a **main** method to test this method, and use the following array as a parameter.

**param\_int** -> [9, 23, -6, 81, -37, 64, 55]

The main method should print the contents of the returned array from the **randomPower** method.

```
import java.util.*;
import java.lang.Math;
class RandomDouble
{
    static int[] randomPower(int[] param_int)
    {
        Random rand = new Random();
        int length = param_int.length;
        int[] random_int = new int[length];
        int[] return_array = new int[length];
        double x;
        double y;

        for(int i=0;i<length;i++)
        {
            random_int[i]=Math.abs(rand.nextInt());
            x=(double)param_int[i];
            y=(double)random_int[i];
            return_array[i] = (int)Math.pow(x,y);
        }
        return return_array;
    } //end randomPower
}
```

```

public static void main(String args[])
{
    int my_int[]={9,23,-6,81,-37,64,55};
    int retArray[] = new int[7];

    retArray=randomPower(my_int);
    for(int i=0;i<7;i++)
    {
        System.out.println(retArray[i]);
    }
}
} //end main

} //end class

```

## 2. (15 points )

Create a class `Transpose`. Within this class, create and initialize a 2D array `myArray` to store a square matrix as shown below:

```

1 2 3 4
5 6 7 8
9 8 7 6
0 4 3 2

```

The transpose of `myArray` is a 2D array called `transposedArray`. It is as shown below:

```

1 5 9 0
2 6 8 4
3 7 7 3
4 8 6 2

```

- a) (12 points) Write a method which takes `myArray` as a parameter and returns the transposed array.

```

import java.util.*;
class Transpose
{

    static int[][] transPosedArray = new int[4][4];
    static int myArray[][]={{1,2,3,4},{5,6,7,8},{9,8,7,6},{0,4,3,2}};

    public static int[][] transpose(int myArray[][])
    {
        int total=4*4;
        int i=0;
        int j=0;
        int tempArray[][]=new int[4][4];
        while(i<4) // You could potentially use a for loop also here
        {
            j=0;
            while(j<4)
            {

```

```

        tempArray[j][i]=myArray[i][j];
        j++;

    }
    i++;
}
return tempArray;
}
}

```

**(b) (3 points)** Explain how a static variable is different from a non-static variable.

A static class variable is any field declared with the static modifier; this tells the compiler that there is exactly one copy of this variable in existence; regardless of how many times the class has been instantiated. Memory space for the static variable is created when the class is first referenced. All objects instantiated from the class share its static class variables. Changing value of static variables in one object will change the value for the variable in all other objects. Non static variables are variables declared without the static keyword. Non-static fields are also known as instance variables because their values are unique to each instance of a class (to each object, in other words).

### **3. (20 points)**

Write a class called **Weather**. It should contain integer class variables **maxTemp**, **minTemp**. The class should also contain a 10 x 7 two-dimensional integer array named **weatherData**. This array will store the weather information of ten cities for seven days. In the constructor for this class, initialize **maxTemp** to **0**, **minTemp** to **70** and each element in the array **weatherData** to a random value between zero and seventy, inclusive.

Create a method called **findMaxMin()**. When this method is called, it should scan through the two-dimensional array and find the maximum temperature and minimum temperature among all the cities during the seven day period. Assign the maximum temperature and minimum temperature to the variables **maxTemp** and **minTemp** respectively. The method should also print out the indices of the city and the day where maximum and minimum temperatures were recorded.

```

import java.util.*;
public class Weather
{
    static int maxTemp;
    static int minTemp;
    static int[][] weatherData = new int[10][7];
    public Weather()
    {
        int i=0;
        int j=0;
        maxTemp=0;
        minTemp=70;
        Random rand = new Random();
        for(i=0;i<10;i++)
        {
            for(j=0;j<7;j++)
            {
                weatherData[i][j] = 0+rand.nextInt(71);
            }
        }
    }
    public static void findMaxMin()
    {
        int temp;
        int maxRow,maxCol,minRow,minCol;
        maxRow=0;
        maxCol=0;
        minRow=0;
        minCol=0;
        for(int i=0;i<10;i++)
        {
            for(int j=0;j<7;j++)
            {
                if(weatherData[i][j]>maxTemp)
                {
                    maxTemp=weatherData[i][j];
                    maxRow=i;
                    maxCol=j;
                }
                if(weatherData[i][j]<minTemp)
                {
                    minTemp=weatherData[i][j];
                    minRow=i;
                    minCol=j;
                }
            }
        }
        System.out.println("Max recorded at City "+maxRow+" on Day "+maxCol);
        System.out.println("Min recorded at City "+minRow+" on Day "+minCol);
    }
}

```

#### 4. (15 points )

Write a class that implements a simple calculator. In the main method, ask the user to enter two integers (step 1). Next, ask the user to choose an operation (step 2). If the user chooses 1, add the two numbers and print the result. If the user chooses 2, multiply the two numbers and print the product. If the user chooses 3, subtract the first number from the second and print the result. If the user chooses 4, divide the first number by the second and print the result. If anything else is entered, the user should be asked to enter an operation once again (e.g. the program should go back to step 2). If a valid operation is entered (e.g. 1 - 4), the method should just print the result and end. Your method should use at least one switch statement. Feel free to create additional methods if you wish.

```
import java.util.*;
class Calculator
{
    public static void main(String args[])
    {
        int num1;
        int num2;
        int choice;
        int flag=1;
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter first integer ");
        num1=scan.nextInt();
        System.out.println("Enter second integer ");
        num2=scan.nextInt();

        while(flag==1)
        {
            System.out.println("1 - Add");
            System.out.println("2 - Multiply");
            System.out.println("3 - Subtract");
            System.out.println("4 - Divide");
            System.out.println("Your choice : ");
            choice=scan.nextInt();
            switch(choice)
            {
                case 1:
                    System.out.println(num1+" + "+num2+" is "+(num1+num2));
                    flag=0;
                    break;
                case 2:
                    System.out.println(num1+"x"+num2+" is "+(num1*num2));
                    flag=0;
                    break;
```

```

    case 3:
        System.out.println(num1+"-"+num2+" is "+(num1-num2));
        flag=0;
        break;
    case 4:
        System.out.println(num1+"/"+num2+" is "+(num1/num2));
        flag=0;
        break;
}
}
}
}

```

### 5. (15 points )

The factorial of a number  $n$ , denoted by  $n!$ , is the product of all positive integers less than or equal to  $n$ .

Example:  $5! = 1 * 2 * 3 * 4 * 5 = 120$ .

Write a method that uses a **for loop** to calculate the factorial of an input integer parameter and print the result. (Hint: You can assume that the input to the method is greater than 0 without checking it)

```

import java.util.*;
class Factorial
{
    public static void factorial(int num)
    {
        int i=0;
        int fact=1;

        for(i=1;i<=num;i++)
        {
            fact=fact*i;
        }
        System.out.println("Factorial is "+fact);
    }
}

```

## 6. (15 points)

In this question you will create a class **StudentTest**. Declare a private **static** integer variable **totalStudents** and a public ArrayList **studentScores** as instance variables within this class. Create *accessor* and *mutator* methods for variable **totalStudents**.

Then, write a main method for the class **StudentTest** which performs the following actions.

- Create an ArrayList object and store the reference in **studentScores**.
- Create five **StudentTest** objects using a loop and initialize the **totalStudents** variable in each to the value of the loop index using the mutator.
- Store the objects in ArrayList **studentScores**.
- Read all objects from **studentScores** and print out the value of **totalStudents** for each object using the accessor.

**Show the output for the main method.**

Hint: This question is VERY similar to the example you coded in discussion last week.

```

import java.util.ArrayList;

class StudentTest
{
    private static int totalStudents;
    public static ArrayList<StudentTest> studentScores;

    int getTotalStudent()
    {
        return totalStudents;
    }

    void setTotalStudent(int val)
    {
        totalStudents=val;
    }

    public static void main(String args[])
    {
        boolean status;
        int x;
        studentScores = new ArrayList<StudentTest>();
        StudentTest st ;

        for(int i=0;i<5;i++)
        {

            st=new StudentTest();
            st.setTotalStudent(i);
            studentScores.add(i,st);
        }

        for(int j=0;j<5;j++)
        {
            st=(StudentTest)studentScores.get(j);
            System.out.println(st.getTotalStudent());
        }
    }
}

```

Output:

4 4 4 4 4