CISC 1115 Fall 2020 Final Exam – Page 1

Name: _____

1.(12 points)
Trace the integer variables in the following code:

```
int x = 5;
int y = 10;
do{
   x++;
   y--;
   while (y >8){
     x = x+2;
     y = y-3;
  }
} while (x < 10);
```

x									
y									

b.

```
for (int i=0; i<5; i=i+2){
  for (int j=i; j>0; j--){ }
}
```

i									
j									

2. (12 points) An integer variable is assigned a value representing inches. Compute and print how many miles, yards, feet and inches that number represents. The rules are: 1 foot equal 12 inches; 36 inches equal 1 yard; 5280 feet equals 1 mile. For example,
245897 inches equals 3 miles 1550 yards 1 foot and 5 inches

3. (6 points) <u>SHOW THE WORK you used to perform the following conversions</u>:

a. 1100110 (base 2) to base 10 Answer _____

b. AB (base 16) to base 10 Answer _____

c. 65 (base 10) to base 2 Answer _____

Name: _____

4. (16 points) For each part below, fill in the table to trace the values as they are passed from main and changed in the method.

a.

```
int [ ] array = {1,2,3,4,5};
int x = 12;
int y = doOne(array,x);


int [ ] array2 = {1,2,3,4,5};
int w = doOne(array2, array2[3]);

} // end of main

public static int doOne(int[] arr, int x){
    for (int i=0; i<arr.length-1; i++)
        arr[i] = arr[i]*arr[i+1];
    x = x+3;
    return x;
}
```

x in main before call to doOne	
array in main before call to doOne	
arr in doOne before loop	
x in doOne before loop	
arr in doOne after loop	
x in doOne after loop	
x in main after call to doOne	
y in main after call to doOne	
array in main after call to doOne	

array2 in main before call to doOne	
arr in doOne before loop	
x in doOne before loop	
arr in doOne after loop	
x in doOne after loop	
w in main after call to doOne	
array2 in main after call to doOne	

Name: _____

b.

```
int []  array3 = {1,2,3,4,5};
int s = 12;
int v = 32;
int [] num = doTwo(array3, v,s);

}  // end of main

public static int[] doTwo(int[] arr, int x, int y){
 int[] ret = new int[2];
 x = arr[2] ;
 y = arr[3] ;
 ret[0] = x+arr[1];
 ret[1] = y-arr[0];
 return ret;
}
```

array3 before call to doTwo	
s before call to doTwo	
v before call to doTwo	
arr in doTwo	
x in doTwo	
y in doTwo	
ret in doTwo	
array3 in main after call to doTwo	
num in main after call to doTwo	
s in main after call to doTwo	
v in main after call to doTwo:	

5. (12 points)

a. Given 6 integers i,j,k,l,m and n all initialized and a method max3 which returns the largest of any 3 numbers, write <u>one</u> Java statement to find the maximum of the 6 integers using any combination of Math.max and max3

b. Write a Java expression that would compute the following

$$\frac{x^2 - \frac{1}{(y+3)}}{\frac{1}{z} + \sqrt{x}}$$

Name: _____

c. Perform a binary search on the following array searching for the number 18

```
int[] nums =  {12, 77, 87, 89, 100, 117, 125, 189, 235, 529, 1000};
```

List the high, low and midpoint values at each step until the algorithm stops. DO NOT write any code. Enter into the table below the low, mid and high values as the binary search progresses. You may use either the subscript (index) values or the actual number stored at the subscript. Not all rows of the table may be needed.

Low	Mid	High

6. (12 points) You are given a string containing a series of 9 digit zip codes with a dash separating the two parts and one space separating each full zip code from the next one. For example,

String str="11230-1234 11011-3489 07621-8845";

Write Java code to print the first 5 digits of each zip code followed by the last 4 digits of the zip code as separate value.

```
For example: 11230 1234
             11011 3489
             07621 8845
```

NOTE: The first part of the full zip code is always 5 digits, the second part is always 4 digits and there's always a dash in between the two parts.

Keep in mind that your code must handle a String that has any number of such pairs, not just the three in the example above

Name: _____

7.(30 points) Write a complete Java program, including at least one comment in the main program and one in each method, to do the following:

The program will read in an unknown number of records from a file. Assume you won't have more than 100 records in total, though the actual number of records can be less than 100. Each record contains a sales rep's first name and miles traveled on two trips per year. For example, *Pauline 167.8 567.0*

Main:

- Declare an output file to be used in main and passed to one or more methods from main. <u>All output from main and methods will be printed to this file.</u>

- Invoke the first method (below) to read the data from the input file, compute averages and store information in the arrays. The method returns the number of records read in.

- Next, invoke the second method (below) which computes the average length of all trips and then prints to the output file how many individual averages are above, equal to and below the overall average.

- Finally, invoke the third method (below) to sort the name and individual average trip length arrays in parallel

- In main, print to the output file each sales rep name and average trip length. The names and average trip length should print in columns, right adjusted. All trip lengths should be printed with two decimal places.

Methods:

1. This method has two parameters: a String array for names and a double array for the average trip length per individual sales rep. It returns an integer
   a. Declare the input file, read the records in the file and store each first name in an array (assume there are no duplicate names) and the average number of miles travelled for that sales rep in another array. You do not need to store the individual trip length.
   b. Return the number of sales rep records read in.

2. This method has three parameters – the output file, double array of individual average trip length and an integer representing the total number of records read in.
   a. Compute the overall average trip length across all sales reps.
   b. Compute and print to the output file how many sales reps individual average trip length were above the overall average trip length, below the overall average trip length and equal to the overall average trip length.

3. This method has three parameters – String array of names, double array of individual average trip length and an integer representing the total number of records read in by the first method.
   a. Sort the sales rep names in descending (reverse) alphabetical order synchronizing the individual sales rep's average trip length in the parallel array.

Name: _____

Questions to Mix and Match

1.

(a) Given an array of strings, write Java code to find the string that would come earliest in the dictionary with the smallest value (Unicode) and its position in the array

(b) Given a string, write Java code to count how many exclamation points there are in the string

2. You will be reading in a series of names from the keyboard. Count how many matching pairs there are – a matching pair is when two consecutive names are equal regardless of their case - and print the pairs. Input stops when the word 'end' is entered (also ignoring case). It's up to you if you use arrays or not.

For example,
Jack Jose jose Faith Robert Cynthia cynthia Cynthia Pierre ENd
would print:

Jose jose
Cynthia cynthia
cynthia Cynthia
There were 3 matching pairs

3. Given an integer, write Java code to print the digits in the reverse order without using arrays.

Name: _____

Q1.

x	5	6	8	9	10				
y	10	9	6	5	4				

i	0	2		4					
j		2	1	4	3	2	1		

}

Q2.

```
int totalInches = 245897;
int miles = totalInches/(5280*12);
int remainder = totalInches%(5280*12);

int yards = remainder/36;
remainder = remainder%36;

int feet = remainder/12;
remainder = remainder%12;
System.out.println
 ("Miles "+miles+" yards "+yards+" feet "+feet+" inches "+remainder);
```

Q3.

a. 102
b. 171
c. 11100111

Q4.

a.

```
x before call to doOne: 12
 array before call to doOne 1 2 3 4 5
 arr in doOne before loop 1 2 3 4 5
 x in doOne before loop : 12
 arr in doOne after loop 2 6 12 20 5
 x in doOne after loop : 15
 x after call to doOne : 12
 y after call to doOney: 15
 array after call to doOne 2 6 12 20 5


 array2 before call 1 2 3 4 5
 arr in doOne before loop 1 2 3 4 5
 x in doOne before loop : 4
 arr in doOne after loop 2 6 12 20 5
 x in doOne after loop : 7
 w after call to doOne : 7
 array2 after call 2 6 12 20 5
```

b.

```
 array3 before call 1 2 3 4 5
 s before call to doTwo : 12
 v before call to doTwo: 32
 arr in doTwo  1 2 3 4 5
 x in doTwo : 32
 y in doTwo  : 12
 ret in doTwo  5 3
 x in doTwo  : 3
 y in doTwo  : 4
 array3 in main after call to doTwo 1 2 3 4 5
 num in main after call to doTwo 5 3
 s in main after call to doTwo : 12
 v in main after call to doTwo: 32
```

Q5.

a.

Name: _____

```
System.out.println(Math.max(max3(i,j,k),max3(l,m,n)));
```

OR

```
System.out.println(max3(Math.max(i,j),Math.max(k,l),Math.max(m,n)));
```

```
b.double w = (x*x-1/(y+3))/(1/z+Math.sqrt(x));
```

c.

Low	Mid	High
12(0)	117(5)	1000(10)
12(0)	87(2)	100(4)
12(0)	12(0)	77(1)
77(1)	77(1)	77(1)
77(1)	NOT FOUND	12(0)

Q6.

```
     int start=0;
     int dash = str.indexOf('-',start);
        while(dash != -1){
            String zip = str.substring(dash-5,dash);
            String zip2 = str.substring(dash+1,dash+5);

            System.out.print(zip+" "+zip2+"\n");

            start=dash+1;
            dash = str.indexOf('-',start);
        }
```

Name: _____

Q7.

```
import java.io.*;
import java.util.*;
public class Main {
    public static void main(String[] args) throws IOException{
        String[] name = new String[100];
        double[] avg = new double[100];
        PrintWriter output = new PrintWriter("c:\\temp\\output.txt");
        int num = readData(output,name,avg);
        computeAndCount(output,avg,num);
        sortArray(name,avg,num);
        for (int i=0;i<num;i++){
            output.printf("\n%10s%7.2f", name[i],avg[i]);
        }
        output.close();
    }
    public static int readData
         (PrintWriter pw, String[] nm, double[] av) throws IOException{
        int count=0;
        double total = 0;

        Scanner input = new Scanner(new File("c:\\temp\\input.txt"));
        while(input.hasNext()){
            nm[count] = input.next();
            double trip1 = input.nextDouble();
            double trip2 = input.nextDouble();
            av[count] = (trip1+trip2)/2.0;
            count++;
        }
        return count;
    }

    public static void computeAndCount (PrintWriter pw, double[] av, int num){
        int countAbove=0, countEqual=0,countBelow=0;

        double overallAvg = 0;
        for (int i=0; i<num;i++)
            overallAvg = overallAvg + av[i];
        overallAvg = overallAvg/num;

        for (int i=0;i<num; i++){
            if (av[i]> overallAvg) countAbove++;
            else if (av[i]<overallAvg)countBelow++;
            else countEqual++;
        }

        pw.println("Number above overall average "+countAbove);
        pw.println("Number equal to overall average "+countEqual);
        pw.println("Number below overall average "+countBelow);
    }

    public static void sortArray (String[] primary, double[] secondary, int k){
        for (int i = 0; i < k; i++)
         {
            for (int j =0; j < k-1-i; j++)
            {
                if ((primary[j].compareTo(primary[j+1]))>0)
                {   String temp = primary[j];
                    primary[j] = primary[j+1];
                    primary[j+1] = temp;
                    double tempD = secondary[j];
                    secondary[j] = secondary[j+1];
                    secondary[j+1] = tempD;
                }
            }
         }
    }
}
```