Name: ______________________________

Write all of your responses on these exam pages. You may use the backs of the pages if needed.

1 Short Answer/Method Construction (15 Points Each)

1. Java is a "platform-independent language." What is a platform, what does platform-independent mean,and how does Java attain its platform independence?

2. Answer the following questions,

   (a) Declare a two-dimensional array of integers that has 125 rows and 17 columns.

   (b) Declare an object named Triangle whose constructor brings in the lengths of the three sides as decimal numbers. When you create the triangle have the sides set to 3.6, 9.5, and 10.7.

   (c) Declare an array list that will hold doubles.

   (d) Name all of the native data types that will cycle around to their most negative values when overloaded.

   (e) Name all of the native data types that will be assigned the value of 0 when underloaded.

3. Write a method that takes in a one-dimensional array of integers and an integer target value. Have the method do a Binary Search of the array to find the target value. If the value is found have the method return the index of the value and if the value is not found have the method return -1.

4. Write a method that takes in a one-dimensional array of integers as its only parameter and sorts it using either the bubble sort, insertion sort, or selection sort.

5. Write a method that will take in a single string parameter and return a string. The return string is to be constructed by taking the parameter string and doing the following to it.

   (a) Remove all non-alphabetic characters from it. Recall that there is a `Character.isAlphabetic(ch)` command that returns true if ch is an alphabetic character and false otherwise.

   (b) Convert all characters to uppercase.

   (c) Put all of the even-indexed characters at the front of the string and all odd-indexed characters at the back.

   For example,

```
String: This is a string method...
Result: TIIATIGEHDHSSSRNMTO
```

6. Write two methods, the first is to be called `populate` which takes in a single integer as its only parameter and returns a one-dimensional array of integers of that size. This array is to be populated with randomly generated 0's and 1's. The second method is to be called `maxRun` which is to take in a one-dimensional integer array as its only parameter and return the length of the longest number of consecutive 1's in the array.

7. Write a method that will take in a single double valued parameter `n` and return a single double value. The method is to declare a double called `r` initialized to 1. Then it is to continually replace `r` with the value of

$$\frac{1}{2}\left(r + \frac{n}{r}\right)$$

until two consecutive values of `r` are within 0.00001 of each other. To determine how far two numbers are apart we simply take the absolute value of the difference between the two numbers. The absolute value can be done with an `if` statement or with the `abs` function inside of the Math class.

2 Program Traces (15 Points Each)

1. For the given input, write the output of the program.

```java
import java.util.Scanner;

public class FinalTrace001 {

    public static void print(int[] A) {
        for (int i = 0; i < A.length; i++)
            System.out.print(A[i] + " ");

        System.out.println();
    }

    public static void copy(int[] A, int[] B)
         {
        if (A.length != B.length)
            return;

        for (int i = 0; i < A.length; i++)
            B[i] = A[i];
    }

    public static int meth1(int[] A) {
        for (int i = 0; i < A.length; i++)
            A[i] = A[i] / 3;

        return A[A.length / 2];
    }

    public static void meth2(int[] A) {
        int temp = A[0];
        for (int i = 0; i < A.length; i++)
            A[i] = A[(i + 1) % A.length];

        A[A.length - 1] = temp;
    }

    public static int[] meth3(int[] A) {
        int[] C = new int[2];

        for (int i = 0; i < A.length; i++)
            if (i % 3 == 0)
                C[0] = A[i];
            else if (i % 4 == 0)
                C[1] = A[i];

        return C;
    }

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System
            .in);
        System.out.print("Input: ");
        int n = keyboard.nextInt();

        int[] A = new int[n];
        int[] B = new int[n];
        for (int i = 0; i < n; i++) {
            int t = keyboard.nextInt();
            if (t < 0)
                t = 0;
            A[i] = t;
        }
        copy(A, B);

        print(A);
        A[A.length - 1] = meth1(A);
        print(A);

        System.out.println("----");

        copy(B, A);
        print(A);
        meth2(A);
        print(A);

        System.out.println("----");

        copy(B, A);
        print(A);
        meth3(A);
        B = meth3(A);
        print(B);
    }
}
```

```
Input: 6 2 17 -5 23 17 10
```

2. For the given input, write the output of the program. For any spaces, including leading or trailing, use an under-bracket ␣ to represent the space. For example, `Hi  There ` should be written as `Hi␣␣There␣`.

```
import java.util.Scanner;

public class FinalTrace002 {

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Input: ");
        String s = keyboard.nextLine();

        s.toUpperCase();
        System.out.println(s);
        String t = s.replaceAll("it", "cat");
        System.out.println(t);
        t = t.replaceAll("is", "lost");
        System.out.println(t);

        int count = 0;
        for (int i = 0; i < s.length(); i++) {
            char c = s.toLowerCase().charAt(i);
            if (c > 'q')
                count++;
        }
        System.out.println(count);
        System.out.println(s);

        t = s.substring(2 * s.length() / 3);
        System.out.println(t);

        t = s.substring(10, s.length() - 10);
        System.out.println(t);

        System.out.println(t.charAt(7));
        System.out.println('Q' - 'F');

        t = "HELP";
        s = "";
        for (int i = 0; i < t.length(); i++) {
            s += (char)(t.charAt(i) + 5);
        }
        System.out.println(s);
    }
}
```

`Input: This␣is␣it,␣I␣have␣had␣it,␣and␣it's␣over.`

3 Coding (25 Points Each)

1. This exercise is to write a program that will simulate the rolling of a single die, store the sequence of rolls in an ArrayList, and then report information on the sequence of rolls. There are to be three methods outside the main program. They are as follows,

   (a) `createArrayList` — This will take in a single integer parameter and return an ArrayList of integers. The integers populating this ArrayList are to be randomly generated between 1 and 6, i.e. a die roll.

   (b) `RunsOfLength` — This will take in two parameters, an ArrayList of integers and an integer n. It will return a single integer which is the count of the number of runs of length n of any value.

   (c) `Tabulate` — This will take in a single parameter, an ArrayList of integers. It will return an array of 6 integers which are the counts of the values of die rolls.

   The main is to ask the user for the number of rolls to do and if they want to print out the ArrayList. It should then call `createArrayList` to construct the ArrayList, print it if requested, call `RunsOfLength` repeatedly to display the numbers of runs of lengths 2-10, call `Tabulate`, then finally print out the tabulation array. All input and screen output must be done in the main and all calculations are to be done in their respective methods. Two runs of the program are below,

```
Input n: 1000000
Print ArrayList (y/n): n

Runs of length 2 = 166833
Runs of length 3 = 27600
Runs of length 4 = 4562
Runs of length 5 = 750
Runs of length 6 = 108
Runs of length 7 = 13
Runs of length 8 = 0
Runs of length 9 = 0
Runs of length 10 = 0

Rolls Count
# of 1's = 166319
# of 2's = 166187
# of 3's = 167178
# of 4's = 166962
# of 5's = 166814
# of 6's = 166540
```

```
Input n: 20
Print ArrayList (y/n): y

[3, 4, 6, 4, 5, 2, 4, 2, 1, 1, 1, 1, 1, 3,
1, 5, 2, 2, 2, 6]

Runs of length 2 = 6
Runs of length 3 = 4
Runs of length 4 = 2
Runs of length 5 = 1
Runs of length 6 = 0
Runs of length 7 = 0
Runs of length 8 = 0
Runs of length 9 = 0
Runs of length 10 = 0

Rolls Count
# of 1's = 6
# of 2's = 5
# of 3's = 2
# of 4's = 3
# of 5's = 2
# of 6's = 2
```

```java
import java.util.ArrayList;
import java.util.Scanner;

public class FinalExam_Prog2_F19 {
```

```
        }
    }
```

2. In this exercise you will be writing a `Rectangle` class and a main that uses it. The rectangle class is to store the height and width of the rectangle and an integer id number for the rectangle. The class should have a constructor that loads in the height, width, and id number as parameters, this constructor will also make sure that the stored height and width are at least 0. A default constructor that sets all values to 0. In addition, this class must have the following six methods,

   (a) `Area` — This returns the area of the rectangle.

   (b) `Perimeter` — This returns the perimeter of the rectangle.

   (c) `Diagonal` — This returns the length of the diagonal of the rectangle.

   (d) `MaxDimension` — This returns the maximum dimension of the rectangle, that is, the larger of the height or the width.

   (e) `AreaGreater` — This returns a boolean that is true if the calling rectangle has a greater area than the parameter rectangle.

   (f) `toString` — Overloaded `toString` method that returns a string of the id number followed by three dashes and then the width and height inside square brackets with a comma between them. For example,

   ```
   562038224 --- [24.83727143187372, 16.16987669761648]
   ```

   The `Rectangle` class is to have no other methods, including no accessor methods. So once the rectangle is constructed no one can change or access the data members of the class.

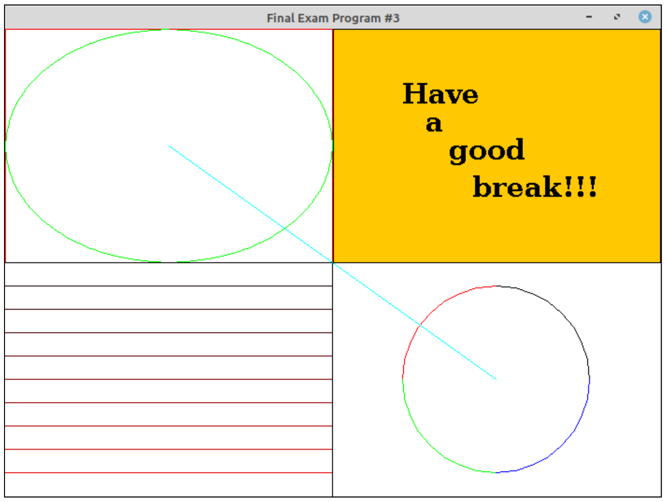
   The main is to ask the user for the number of rectangles that they want, create an array that stores that many rectangles, populate the array with random rectangles who's height and width are in the range from 0 to 100 and the ID is a random integer from 0 to 1,000,000,000. Have the main print out the ID, height, width, and area of each rectangle in the array, in the format of the example below. Create a method that will sort the rectangle array from smallest to largest by the area of the rectangle. Have the main then print the sorted array out in the same format. Create a method to print a single rectangle's information in the format in the example below. Have the main call this method with the first rectangle of the sorted array. An example run is below.

```
Input n: 5

179164914 --- [64.2956126195429, 50.95911558467422]: 3276.447555066725
53268840 --- [91.74299930608926, 43.67708058930935]: 4007.066374197012
562038224 --- [24.83727143187372, 16.16987669761648]: 401.6156165586304
646575810 --- [57.83792869751343, 25.25651944988063]: 1460.7847710895571
959789073 --- [64.51060284253573, 4.878251354197882]: 314.6989356767217

959789073 --- [64.51060284253573, 4.878251354197882]: 314.6989356767217
562038224 --- [24.83727143187372, 16.16987669761648]: 401.6156165586304
646575810 --- [57.83792869751343, 25.25651944988063]: 1460.7847710895571
179164914 --- [64.2956126195429, 50.95911558467422]: 3276.447555066725
53268840 --- [91.74299930608926, 43.67708058930935]: 4007.066374197012

959789073 --- [64.51060284253573, 4.878251354197882]
Area = 314.6989356767217
Perimeter = 138.77770839346724
Diagonal = 64.69478507099403
Maximum Dimension = 64.51060284253573
```

Write the code for the Rectangle class here.

Write the code for the main program and support methods here.

3. Write the paint method code for the GraphicsJPanel to create the following image. The GraphicsJPanel has a height of 500 pixels and width of 700 pixels. The font is Serif, bold, and size 30.



```java
public void paint(Graphics g) {
    super.paint(g);
}
```