1 Short Answer/Method Creation (10 Points Each)

1. Write the following declarations:
   (a) An integer with a random value between -17 and 39.
   (b) An array list that stores doubles.
   (c) A two dimensional array of longs with 29 rows and 127 columns.
   (d) A double with a random value between -10 and 20.
   (e) A triangle with side lengths of 4, 6, and 7. The object class has name Triangle and the constructor takes in the three side lengths.

2. (a) What happens when you overload or underload an integer?
   (b) What happens when you overload or underload a double?

3. What are the three types of programming errors? Briefly describe each of them.

4. Write a method that calculates the factorial of an integer parameter n. The method should return a long. Recall that the factorial is defined as $0! = 1$ and if $n > 0$ then $n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$. Also have the method return -1 if the value of n is less than 0.

```
public static long Factorial(int n) {
    if (n < 0)
        return -1;
    else if (n == 0)
        return 1;
    else {
        long fact = 1;
        for (int i = 1; i <= n; i++)
            fact*= i;

        return fact;
    }
}
```

5. Write a method that will simulate the rolling of 20 six-sided dice and return the number of rolls needed for all 20 dice to be equal.

```
public static long Rolls() {
    int A[] = new int[20];
    long roll = 0;
    boolean done = false;

    do {
        for (int i = 0; i < A.length; i++)
            A[i] = (int) (Math.random() * 6) + 1;

        done = true;
        for (int i = 0; i < A.length; i++)
            if (A[i] != A[0])
                done = false;

        roll++;
    } while (!done);

    return roll;
}
```

6. Write a method that will take in two string parameters and return a new string that is the first string with the last occurrence of the second string removed. For example, if the two parameter strings are the first two lines then the output of the method is the third line.

```
Not␣all␣languages␣have␣compilers.␣Some␣languages␣are␣interpreted.
er
Not␣all␣languages␣have␣compilers.␣Some␣languages␣are␣intpreted.
```

```
public static String RemoveLast(String str1, String str2) {
    int pos = str1.lastIndexOf(str2);
    if (pos == -1)
        return str1;
    else
        return str1.substring(0, pos) + str1.substring(pos + str2.length());
}
```

7. Write a linear search method for an integer array that takes in an array and target value as parameters and returns the first position of the target in the array. If the target is not in the array then the method should return -1;

```
public static int search(int A[], int t) {
    for (int i = 0; i < A.length; i++)
        if (A[i] == t)
            return i;

    return -1;
}
```

8. Write a method that will take in a one-dimensional array and shift all of the entries to the right with the last entry cycling around to the first. For example, if the original array is on the first line then the array after the method call is the second line.

```
5    6    3    6    1
1    5    6    3    6
```

```
public static void Shift(int A[]) {
    int last = A[A.length - 1];

    for (int i = A.length - 1; i > 0; i--)
        A[i] = A[i - 1];

    A[0] = last;
}
```

9. Write a method that takes in a two dimensional array of doubles and returns a one dimensional array of column sums.

```
public static double[] ColSums(double A[][]) {
    double B[] = new double[A[0].length];

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

    return B;
}
```

10. Write a method that takes in a one-dimensional array of integers as its only parameter and sorts it using the bubble sort.

```
public static void BubbleSort(int A[]) {
    for (int i = A.length - 1; i > 0; i--) {
        for (int j = 0; j < i; j++) {
            if (A[j] > A[j + 1]) {
                int temp = A[j];
                A[j] = A[j + 1];
                A[j + 1] = temp;
            }
        }
    }
}
```

2 Program Trace (20 Points Each)

1. For each input show the program output.

```
import java.util.Scanner;

public class FinalTrace001 {

    public static int Method3(int n1, int n2) {
        System.out.println("In Method 3");
        int b = 0;
        int c = 5;
        n2 = n1 + n2;
        for (int i = 0; i < n2; i++)
            b += c;
        System.out.println(b);
        return b;
    }

    public static int Method2(int n1, int n2) {
        System.out.println("In Method 2");
        int a = 1;
        n2 = n1 - n2;
        for (int i = 0; i < n2; i++)
            a += a;
        System.out.println(a);
        return a;
    }

    public static int Method1(int n1, int n2,
            int n3) {
        System.out.println("In Method 1");
        if (n1 > n2)
            return 4;
        else if (n3 > n2)
            return Method2(n2, n1);
        else
            return Method3(n1, Method2(n2 / 5,
                n3 / 3));
    }

    public static void print(int n1, int n2,
            int n3) {
        System.out.println(n1 + " " + n2 +
            " " + n3);
    }

    public static void print(int n1, int n2,
            int n3, int n4) {
        System.out.println("#" + n1 + " " +
            n2 + " " + n3 + " " + n4);
    }

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int a, b, c;
        System.out.print("Input Number 1: ");
        int n1 = keyboard.nextInt();
        System.out.print("Input Number 2: ");
        int n2 = keyboard.nextInt();
        System.out.print("Input Number 3: ");
        int n3 = keyboard.nextInt();
        a = n3;
        c = n2;
        b = a - c;
        while (a > b) {
            b += n1;
            c = a - b;
            a--;
            print(a, b, c);
        }
        print(a, b, c);
        int t = Method2(a, b);
        print(a, b, t);
        t = Method3(c, b);
        print(c, b, t);
        t = Method1(a, b, c);
        print(a, b, c, t);
    }
}
```

(a)
```
Input Number 1: 5
Input Number 2: 3
Input Number 3: 1

0 3 -2
0 3 -2
In Method 2
1
0 3 1
In Method 3
5
-2 3 5
In Method 1
In Method 2
1
In Method 3
5
#0 3 -2 5
```

(b)
```
Input Number 1: 3
Input Number 2: 9
Input Number 3: 6

5 0 6
4 3 2
3 6 -2
3 6 -2
In Method 2
1
3 6 1
In Method 3
20
-2 6 20
In Method 1
In Method 2
2
In Method 3
25
#3 6 -2 25
```

2. For the following input show the program output.

```
import java.util.Scanner;

public class FinalTrace002 {

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

        System.out.println();
    }

    public static void reload(int A[]) {
        for (int i = 0; i < A.length; i++)
            A[i] = 2 * (i + 1);
    }

    public static void Method1(int A[]) {
        for (int i = 0; i < A.length; i++)
            if (A[i] > 10)
                A[i] -= 10;

        for (int i = 0; i < A.length; i++)
            if (A[i] < 5)
                A[i] += 5;
    }

    public static void Method2(int A[]) {
        for (int i = 0; i < A.length / 2; i++)
            A[i] = A[A.length - 1 - i];
    }

    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];
        reload(A);
        print(A);
        for (int i = 0; i < A.length; i++)
            A[i] = A[(4 * i + 1) % n];
        print(A);
        Method1(A);
        print(A);
        System.out.println("-----");
        reload(A);
        for (int i = A.length - 1; i > 1; i--)
            A[i] = A[i] / A[i - 1];
        print(A);
        System.out.println("-----");
        reload(A);
        Method2(A);
        print(A);
        Method1(A);
        print(A);
        System.out.println("-----");
        reload(A);
        B = A;
        for (int i = 0; i < A.length; i++)
            A[i] += B[i];
        Method1(A);
        Method2(B);
        print(A);
        print(B);
    }
}
```

Input: 7

```
2 4 6 8 10 12 14
4 12 6 14 14 4 14
9 7 6 9 9 9 9
-----
2 4 1 1 1 1 1
-----
14 12 10 8 10 12 14
9 7 10 8 10 7 9
-----
18 14 10 6 10 14 18
18 14 10 6 10 14 18
```

3 Coding (25 Points Each)

1. Write a program that asks the user for a string (possibly several words) and have the program print out the number of letters, the number of spaces, and the number of words. Note that two words may have more than one space between them.

2. Write a program that will take integer input from the user until the user types in a negative number. For each input the program is to print out if the number is even or odd. Once the user types in a negative number the program is to display the minimum input and the maximum input. You may not use an array or ArrayList for this program.

3. Create a `Pyramid` class that stores three double values for the length of the base, the width of the base, and the height of the pyramid. The class is to have a single constructor that loads in the length, width, and height into the object. The class is to have three more methods,

   - `Volume` — This returns the volume of the pyramid. The volume of a pyramid is $\frac{1}{3}$ the area of the base times the height.
   - `SurfaceArea` — This returns the surface area of the pyramid. The surface area of a pyramid is the sum of the areas of the 5 faces. That is, the rectangular base and the four triangles. Recall that the area of a triangle can be found using Heron's formula. If the lengths of the sides of a triangle are a, b, and c then the area of the triangle is $A = \sqrt{p(p-a)(p-b)(p-c)}$, where p is the semi-perimeter, that is, $p = \frac{a+b+c}{2}$.
   - `EdgeLength` — This returns the edge length of the pyramid. Which is the sum of the lengths of the 8 edges of the pyramid.

   To do these calculations, you will need the length of the side s in the picture below. The length s is called the slant height of the pyramid. If L, W, and H are the length, width, and height of the pyramid then the slant height is, $s = \sqrt{H^2 + \left(\frac{L}{2}\right)^2 + \left(\frac{W}{2}\right)^2}$.

   Also write a main program that creates a pyramid with length 5, width 3 and height 7, then prints out the volume, surface area, and edge length.