Name: _____

Write all of your responses on these exam pages.

1 Short Answer (5 Points Each)

1. What is the difference between a compiler and an interpreter? Also, discuss Java's method.

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

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

4. What are reserved words? Give four examples of Java reserved words.

5. Answer the following questions about numeric data types in Java.

   (a) What happens when you overload an int?

   (b) What happens when you overload a double?

   (c) What happens when you underload an int?

   (d) What happens when you underload a double?

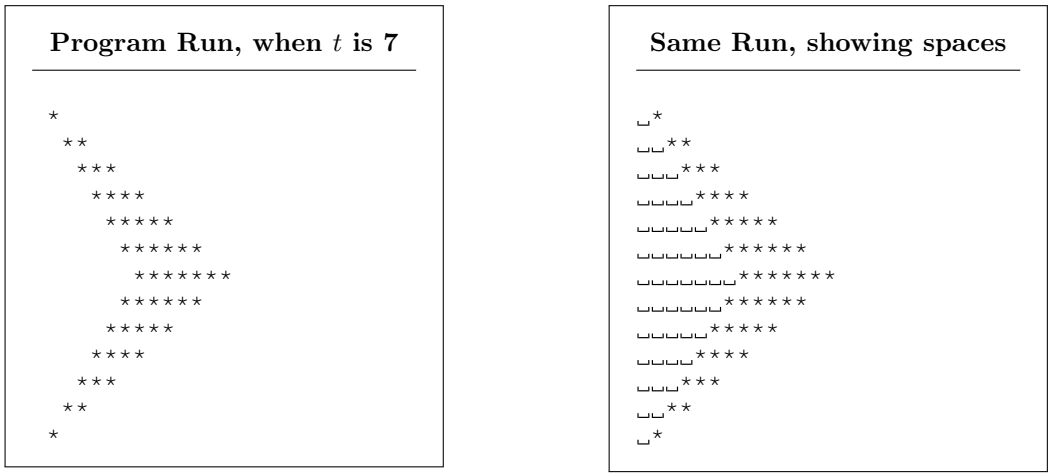
   (e) What happens when you input an integer into position 10 of an integer array of size 10?

2 Short Coding Answer (10 Points Each)

1. Write a segment of code that will simulate the rolling of two die and counts the number of rolls it takes to get snake-eyes 5 times in a row. Snake-eyes is when both die have a 1 on the upper face.

2. Write a function called `Factorial` that takes a long as a parameter and returns the factorial of that long, the return type is also a long. Recall that the definition of a factorial is $n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1$ and we define $0! = 1$.

3. Write a segment of code that takes an integer value of `t`, which you are to assume has a positive value before the segment of code, and then produces the following image.



In the image the number of spaces before the $\star$ are the same as the number of $\star$'s. These start at one and go up to the value of t, and then back down to 1. In addition, you may only use the following print statements in the code.

```
System.out.print(" ");
System.out.print("*");
System.out.println();
```

4. 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.

5. Write a method that does either the bubble sort, insertion sort or selection sort for an array of integers. You must state which sort you are writing.

6. Write a segment of code that will add up all of the entries on the diagonal of a two-dimensional integer array, `A`. The diagonal of an array are the entries in the $(0,0)$, $(1,1)$, $(2,2)$, $\ldots$, (n,n) positions, where the (n,n) position is the largest possible position that is still in the array. So for a 4×9 array $n = 4$ and for a 5×3 array $n = 3$. You do not know the size of the array before this block of code is executed.

7. Write a segment of code that will declare an ArrayList of integers, populate the list with the numbers from 1 to 1000000, then continually remove the last entry until only 10 entries remain in the list, then finally print out the list.

8. Write a segment of code that will take an integer array `A` and sum up all of the even indexed entries and subtract off all of the odd indexed entries.

3 Program Traces (20 Points Each)

1. For each of the three runs, give the program output.

```
1  import java.util.Scanner;
2
3  public class FinalTrace1 {
4
5      public static void main(String[] args) {
6          Scanner keyboard = new Scanner(System.in);
7          System.out.print("n = ");
8          int n = keyboard.nextInt();
9          System.out.print("m = ");
10         int m = keyboard.nextInt();
11
12         int j = 1;
13         int i = 10;
14
15         do {
16             System.out.println("n is " + n);
17             if (n % 3 == 0) {
18                 j++;
19             } else if (n % 3 == 1) {
20                 i = i - 3 * j;
21             } else {
22                 i++;
23                 j++;
24             }
25             n++;
26             m -= 2;
27         } while (n < m);
28
29         System.out.println("i is " + i);
30         System.out.println("j is " + j);
31     }
32 }
```

Run #1

```
n = 3
m = 16
```

Run #2

```
n = 6
m = 6
```

Run #3

```
n = 5
m = 21
```

2. Give the program output.

```
1  import java.util.Scanner;
2
3  public class FinalTrace2 {
4
5      public static int doSomething(int a) {
6          return (int) Math.pow(a, 2);
7      }
8
9      public static int doSomething(int a, int b) {
10         return a / b;
11     }
12
13     public static String doSomething(int a, int b,
           String c) {
14         if (a > b) {
15             int t = a;
16             a = b;
17             b = t;
18         }
19         return c.substring(a, b);
20     }
21
22     public static void main(String[] args) {
23         Scanner keyboard = new Scanner(System.in);
24         System.out.print("s = ");
25         String s = keyboard.nextLine();
26         System.out.print("n = ");
27         int n = keyboard.nextInt();
28         System.out.print("m = ");
29         int m = keyboard.nextInt();
30         System.out.println("-----");
31
32         System.out.println(s.length());
33         System.out.println(doSomething(n, m));
34         System.out.println(doSomething(n, m, s));
35
36         System.out.println("-----");
37
38         int t = doSomething(n) + s.length() / 3;
39         int r = m / 5;
40         System.out.println(t);
41         System.out.println(r);
42         System.out.println(doSomething(t, r, s));
43
44         System.out.println("-----");
45
46         System.out.println(doSomething(n));
47         System.out.println(doSomething(doSomething(
               n)));
48         System.out.println(doSomething(doSomething(
               doSomething(n)), m));
49
50         System.out.println("-----");
51
52         System.out.println(doSomething(m, n));
53         System.out.println(2 * (doSomething(n) - 1)
               );
54         System.out.println(doSomething(doSomething(
               m, n), 2 * (doSomething(n) - 1), s));
55     }
56 }
```

Program Run

```
s = This is a test of the program.
n = 3
m = 11
```

3. For each of the three runs, give the program output.

```
1  import java.util.ArrayList;
2  import java.util.Scanner;
3
4  public class FinalTrace3 {
5
6      public static void print(int M[]) {
7          for (int i = 0; i < M.length; i++)
8              System.out.printf("%3d", M[i]);
9          System.out.println();
10     }
11
12     public static void move(int M[]) {
13         for (int i = 0; i < M.length; i++)
14             M[i] = M[2 * i % M.length];
15     }
16
17     public static void main(String[] args) {
18         Scanner keyboard = new Scanner(System.in);
19         System.out.print("n = ");
20         int n = keyboard.nextInt();
21
22         if (n < 0)
23             n = -n;
24
25         if (n == 0)
26             n = 10;
27
28         int[] A = new int[n];
29         int[] B = new int[n / 3];
30
31         for (int i = 0; i < n; i++) {
32             A[i] = i + 1;
33         }
34
35         print(A);
36         print(B);
37         System.out.println("-----");
38
39         for (int i = 0; i < B.length; i++) {
40             B[i] = A[2 * i % A.length];
41         }
42
43         print(A);
44         print(B);
45         System.out.println("-----");
46
47         move(A);
48         print(A);
49         System.out.println("-----");
50
51         A = B;
52         B[0] = 25;
53         A[A.length - 1] = -7;
54
55         print(A);
56         print(B);
57     }
58 }
```

Run #1

```
n = 5
```

Run #2

```
n = 10
```

Run #3

```
n = 12
```

4 Coding (20 Points Each)

1. Write a program that will take integer input from the user until the user types in a negative number. You may assume that the user does not type in a number larger than 1,000,000,000. 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. A sample run is on the right.

Program

```
import java.util.Scanner;

public class FinalCode001 {

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);









}
```

Sample Run

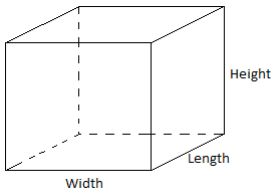
```
Input: 5
Input was odd.
Input: 24
Input was even.
Input: 9
Input was odd.
Input: 2
Input was even.
Input: 7
Input was odd.
Input: 15
Input was odd.
Input: -6
Input was even.

Minimum = 2
Maximum = 24
```

2. Create a `Box` class that stores three double values for the length, width, and height of a box. 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 box.
   - `surfaceArea` — This returns the surface area of the box. Which is the sum of the areas of the 6 faces.
   - `edgeLength` — This returns the edge length of the box. Which is the sum of the lengths of the 12 edges of the box.

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



Main Program

Box Class

3. The main program and output from the program are below. Write the five methods for the program.

   - `print` — The print method is to take in as a parameter only a single two-dimensional array of integers, and print the contents of the array in columns that line up and use 6 spaces for each entry.
   - `populate` — The populate method is to take in as a parameter only a single two-dimensional array of integers, and fill the array with random integers between -50 and 50.
   - `Add` — The Add method is to take as parameters two, two-dimensional arrays and return a two-dimensional array that is the same size as the two input arrays and is the sum of the two arrays. The sum of two arrays is the array where each corresponding cell is added. So the $(0, 0)$ positions of the two are added, the $(0, 1)$ positions of the two are added, the $(1, 1)$ positions of the two are added, and so on. If the sizes of the two input arrays are not the same then they cannot be added and in this case the method should return `null`.
   - `Subtract` — The Subtract method is to take as parameters two, two-dimensional arrays and return a two-dimensional array that is the same size as the two input arrays and is the difference of the two arrays. The difference of two arrays is the array where each corresponding cell is subtracted. So the $(0, 0)$ positions of the two are subtracted, the $(0, 1)$ positions of the two are subtracted, the $(1, 1)$ positions of the two are subtracted, and so on. If the sizes of the two input arrays are not the same then they cannot be subtracted and in this case the method should return `null`.
   - `transpose` — The transpose method is to take in as a parameter only a single two-dimensional array of integers, and return another two-dimensional array that transposes the array. The transpose of an array is where each row becomes a column and each column a row. That is, the first row of the original array is the first column of the transpose, and so on.

   The populate method is the only method that is allowed to alter the input array.

```
1  public static void main(String[] args) {
2      Scanner keyboard = new Scanner(System.in);
3      System.out.print("Rows: ");
4      int rows = keyboard.nextInt();
5      System.out.print("Columns: ");
6      int cols = keyboard.nextInt();
7
8      int[][] A = new int[rows][cols];
9      int[][] B = new int[rows][cols];
10     populate(A);
11     populate(B);
12
13     print(A);
14     System.out.println();
15     print(B);
16
17     int[][] sum = Add(A, B);
18     if (sum != null) {
19         System.out.println();
20         print(sum);
21     }
22
23     int[][] difference = Subtract(A, B);
24     if (difference != null) {
25         System.out.println();
26         print(difference);
27     }
28
29     System.out.println();
30     print(transpose(A));
31 }
```

Sample Run

```
Rows: 3
Columns: 4
   -29    11    48    44
   -35    15    20   -37
   -15     6    18    43

    11     1    45    39
    12    32    50   -29
   -24   -41   -38    17

   -18    12    93    83
   -23    47    70   -66
   -39   -35   -20    60

   -40    10     3     5
   -47   -17   -30    -8
     9    47    56    26

   -29   -35   -15
    11    15     6
    48    20    18
    44   -37    43
```

print method

populate method

Add method

Subtract method

transpose method