Name: _____

Write all of your responses on these exam pages. If you need extra space please use the backs of the pages.

1 Short Answer & Methods (10 Points Each)

1. Write the following Java declarations.

   (a) Declare a decimal number named `piApprox` and set its value to 3.14159.

   (b) Declare a long integer named `national_debt` and set its value to 21,000,000,000,000.

   (c) Declare a one-dimensional array of integers that has 1000 entries in it.

   (d) Declare a two-dimensional array of doubles that has 25 rows and 42 columns.

   (e) 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 1.5, 2.3, and 2,7.

2. What is the difference between a compiler and an interpreter? Also, discuss Java's method. In addition, explain why this makes Java a "platform-independent language."

3. What is method/function overloading? How is it done in Java?

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

5. Write a method that will take as a parameter the number of rolls to do, simulate the rolling of two die that many times, and return the number of "snake eyes" that were rolled. Snake eyes is when both die have a value of 1.

6. Write a method that will take two parameters, both integers, the number of points a student has earned in a class and the total number of points that are possible for the class. Have the program print out to the screen the student's final percentage average and letter grade based on the 90-80-70-60 scale.

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.

8. Write a segment of code that creates an ArrayList of integers, loads in the numbers, 23, 43, 65, 12, 11, 103, 55, and 10, in that order, prints the contents of the ArrayList, then sorts the list in ascending order, and finally prints the ArrayList out again. Do this in as few lines of code as possible. You may assume that all necessary imports are done.

9. Write a method called `DoubleFactorial` that takes a long as a parameter and returns the double factorial of that long, the return type is also a long. Recall that the definition of a double factorial is $n!! = n \cdot (n-2) \cdot (n-4) \cdots 1$ if $n \geq 1$ and $0!! = 1$

10. Write a method that will take as input a single parameter of a two-dimensional array of integers and return a one-dimensional of doubles which holds the row averages of the two-dimensional array.

2 Trace (20 Points)

Write the output of the following program for the given inputs.

```
1  import java.util.Scanner;
2
3  public class FinalTrace001 {
4
5      public static int Method3(int n1, int n2) {
6          System.out.println("In Method 3");
7          int retval = n1 * n2;
8
9          if (retval > 15)
10             retval--;
11         else
12             retval += 5;
13
14         System.out.println("Method 3 Return: " +
               retval);
15         return retval;
16     }
17
18     public static int Method2(int n1, int n2) {
19         System.out.println("In Method 2");
20         int retval = n1 / n2;
21         int retval2 = n1 % n2;
22
23         if (retval > retval2)
24             retval = retval2;
25
26         System.out.println("Method 2 Return: " +
               retval);
27         return retval;
28     }
29
30     public static int Method1(int n1, int n2, int n3)
           {
31         System.out.println("In Method 1");
32         if (n1 > n2)
33             return Method2(n3, n2);
34         else if (n3 > n2)
35             return Method3(n2, n1);
36         else
37             return Method3(n1, Method2(n2, n3));
38     }
39
40     public static void main(String[] args) {
41         Scanner keyboard = new Scanner(System.in);
42         System.out.print("Input Number 1: ");
43         int n1 = keyboard.nextInt();
44         System.out.print("Input Number 2: ");
45         int n2 = keyboard.nextInt();
46         System.out.print("Input Number 3: ");
47         int n3 = keyboard.nextInt();
48
49         System.out.println(Method1(n1, n2, n3));
50     }
51 }
```

1. ```
   Input Number 1: 3
   Input Number 2: 12
   Input Number 3: 11
   ```

2. ```
   Input Number 1: 27
   Input Number 2: 5
   Input Number 3: 13
   ```

3 Coding (30 Points Each)

1. Write a method that will randomly roll 5 dice and test to see if the dice are all the same value. If they are all the same value then have the method return true and if they are not all the same value have the method return false. Write a main program that will continually call this method and count the number of times the rolls must be done until the dice are all the same. Have the main print out this number. You may assume that the following code is given.

```
import java.util.Scanner;

public class DiceSim {

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

2. Write a program that will simulate the throwing of darts on a dart board to estimate the value of π. Specifically, the program should take the number of throws to do from the user. Simulate throwing that many darts at a board that is one unit in diameter which is centered on a square two units on each edge. Take the number of throws that land inside the board divided by the total number of throws and multiplies that value by 4. Have the program print the approximation of π to the screen. You may assume that the following code is given.

```
import java.util.Scanner;

public class PiApprox {

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

3. 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 and do error checking so that if a negative number is input for any of the lengths a 0 is stored instead. The class is to have three more methods,

   (a) **Volume** — This returns the volume of the box.

   (b) **SurfaceArea** — This returns the surface area of the box. Note that the surface area of a box is the sum of the rectangle areas of the 6 sides (or faces) of the box.

   (c) **EdgeLength** — This returns the edge length of the box. Which is the sum of the lengths of the 12 edges of the pyramid.