1 Short Answer (5 Points Each)

1. What is a sentinel value.

   Solution: Special value or values that control program flow, usually used with loops to stop the repetition. Normally it is a value that is either input or changed inside the body of a loop.

2. What are the three types of loops? For each type state if they are precondition or postcondition and what type of control they use.

   Solution:

Type	Check	Control
While	Precondition	Conditional
Do While	Postcondition	Conditional
For	Precondition	Count

3. What is priming a loop?

   Solution: When using a precondition loop it is setting the condition so that you are guaranteed to enter the loop the first time. This usually means that a postcondition loop would be more appropriate.

4. Write a method named `CircleArea` that will bring in a single decimal parameter named `radius` and return the area of a circle with that radius.

   Solution:

```
public static double CircleArea(double radius) {
    return Math.PI * radius * radius;
}
```

5. Write a method named `NiftySequence` that will bring in a single integer parameter named `n` and return $\frac{n}{2}$ if n is even and $3n + 1$ if n is odd.

   Solution:

```
public static int NiftySequence(int n) {
    if (n % 2 == 0) {
        n = n / 2;
    } else {
        n = 3 * n + 1;
    }
    return n;
}
```

6. Write a method named `maximum` that will bring in three decimal parameters and return the maximum of the values.

   Solution:

```
public static double maximum(double a, double b, double c) {
    double max = a;
    if (b > max)
        max = b;
    if (c > max)
        max = c;

    return max;
}
```

2 Program Traces (7 Points Each)

For each of the following segments of code write the output of the program.

1.

```
1  int i = 1;
2  for (int k = 0; k < 5; k++) {
3      for (int j = 1; j < k; j++) {
4          i = i + j;
5      }
6      System.out.println(i);
7  }
```

Solution:

```
1
1
2
5
11
```

2.

```
1  int t = 3;
2  int q = 15;
3  while (q > t) {
4      if (t > 5) {
5          t--;
6          q = 2 * q / 3;
7      }
8      System.out.println(q + "  " + t);
9      t += q / 4;
10     q--;
11 }
```

Solution:

```
15  3
9  5
5  6
```

3.

```
1  public static int doThis(int a, int b, int c) {
2      return a + b / c;
3  }
4
5  public static void main(String[] args) {
6      int b = 4;
7      int c = 7;
8      for (int a = 3; a < 7; a++) {
9          b = doThis(b, c, a);
10         c = doThis(c, b, a);
11         System.out.println(a + " " + b + " " + c);
12     }
13 }
```

Solution:

```
3 6 9
4 8 11
5 10 13
6 12 15
```

4.

```
1  String s = "";
2  for (int k = -1; k < 5; k++) {
3      s = "*" + s;
4      System.out.println(s);
5  }
6
7  for (int k = 2; k < 8; k++) {
8      s = s.substring(1);
9      System.out.println(s);
10 }
```

Solution:

```
*
**
***
****
*****
******
*****
****
***
**
*
␣
```

3 Coding (25 Points Each)

1. Write a program that will take numeric values from the user until the user inputs negative number. Assume the user will type in values between 0 and 1,000,000,000 until they input a negative number to stop the process. The program should then report the average, minimum, and maximum of the values input. A run of the program is below.

```
Input a number from 0 to 1000000000 (< 0 to quit): 3.5
Input a number from 0 to 1000000000 (< 0 to quit): 7.9
Input a number from 0 to 1000000000 (< 0 to quit): 15.6
Input a number from 0 to 1000000000 (< 0 to quit): 0.4
Input a number from 0 to 1000000000 (< 0 to quit): 6.3
Input a number from 0 to 1000000000 (< 0 to quit): -1

Maximum = 15.6
Minimum = 0.4
Average = 6.739999999999999
```

Solution:

```
1  import java.util.Scanner;
2
3  public class Exam02Prog01 {
4
5      public static void main(String[] args) {
6          Scanner keyboard = new Scanner(System.in);
7
8          int value = 0;
9          int sum = 0;
10         int count = 0;
11         int max = 0;
12         int min = 1000000000;
13         while (value >= 0) {
14             System.out.print("Input a number from 0 to 1000000000 (< 0 to quit): ");
15             value = keyboard.nextInt();
16
17             if (value >= 0) {
18                 sum += value;
19                 count++;
20
21                 if (value < min)
22                     min = value;
23
24                 if (value > max)
25                     max = value;
26             }
27         }
28
29         System.out.println();
30         System.out.println("Maximum = " + max);
31         System.out.println("Minimum = " + min);
32         System.out.println("Average = " + 1.0 * sum / count);
33     }
34 }
```

2. Write a program that will ask the user for the number of dice rolls they would like to do. The program should then roll three dice that many times. The three dice are all different in the number of sides (faces)they have. One die is the regular 6 sided die with values between 1 and 6, one die is a 20 sided die with values between 1 and 20, and the third die is a 4 sided die with values between 1 and 4. Each time the program rolls the dice it will take the sum of the face values of the roll. It will count the number of rolls that are even and the number of rolls that are odd. At the end it will display the two counts. The dice rolls and the sum calculation will be in a method called `rollDice` which will roll the dice and return the sum of the roll back to the main. The main will do the rest. A run of the program is below.

```
Input number of rolls: 100000
The number of even rolls was 50146
The number of odd rolls was 49854
```

Solution:

```java
import java.util.Scanner;

public class Exam02Prog02 {

    public static int rollDice() {
        int roll1 = (int) (Math.random() * 6) + 1;
        int roll2 = (int) (Math.random() * 20) + 1;
        int roll3 = (int) (Math.random() * 4) + 1;

        int sum = roll1 + roll2 + roll3;

        return sum;
    }

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

        System.out.print("Input number of rolls: ");
        int rolls = keyboard.nextInt();
        int evenrolls = 0;
        int oddrolls = 0;

        for (int i = 0; i < rolls; i++) {
            int sum = rollDice();

            if (sum % 2 == 0)
                evenrolls++;
            else
                oddrolls++;
        }
        System.out.println("The number of even rolls was " + evenrolls);
        System.out.println("The number of odd rolls was " + oddrolls);
    }
}
```