1 Short Answer (5 Points Each)

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

   Solution:

   Syntax Error: An error in the program code due to misuse of the programming language.

   Run-time Error: An error that occurs during a run of the program which usually causes the program to terminate prematurely.

   Logic Error: This error occurs when the program is syntactically correct and there are no runtime errors but the program does not do what it was intended to do.

2. Write a single line of code (not an entire program) that declares an integer variable `num1` and assigns to it a random integer between -10 and 37, inclusively.

   Solution:

   ```
   int num1 = (int) (Math.random() * 48) - 10;
   ```

3. Write a few lines of code (not an entire program) that will take an input string from the user (an entire line of text) and store it in the variable `str1`, remove all leading and trailing spaces (but no other spaces), extract the first word and then convert that word to all uppercase characters, reassign `str1` to this word, and finally print `str1` out to the screen. You may assume that a Scanner object has been created with name `kb`.

   Solution:

   ```
   String str1 = kb.nextLine();
   str1 = str1.trim();
   int space = str1.indexOf(" ");
   String word = str1.substring(0, space);
   word = word.toUpperCase();
   str1 = word;
   System.out.println(str1);
   ```

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

   (a) What happens when you overload an int?
   Solution: The value cycles around to the minimum value of an int.

   (b) What happens when you overload a double?
   Solution: The value turns into Infinity.

   (c) What happens when you underload an int?
   Solution: The value cycles around to the maximum value of an int.

   (d) What happens when you underload a double?
   Solution: The value turns into 0.

   (e) What happens when you input a non-numeric string when the Scanner is doing a nextInt?
   Solution: Run-time error, since it will not automatically convert a string to an int.

5. 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."

   Solution:

   A compiler will take a program written in a high-level language, translate it into machine language and then save the machine language program to a file that can be run on the computer. An interpreter does essentially the same thing except that it translates the high-level language to machine language one command at a time and does not save the machine language program to a file. Java uses a combination of the two. There is a compile stage that translates the Java code into byte-code that the interpreter (known as the JVM or Java Virtual Machine) runs.

   Java is compiled into byte-code, this byte code is then interpreted by the Java Virtual Machine (JVM). There is a JVM built for every common operating system, so Java byte-code can be run on any operating system.

6. Write a few lines of code that will take in three integers as input, you may assume that a Scanner object has been created with name `kb`, and print out the largest, smallest, and average of the numbers.

   Solution:

```
System.out.print("Input x: ");
int x = kb.nextInt();
System.out.print("Input y: ");
int y = kb.nextInt();
System.out.print("Input z: ");
int z = kb.nextInt();

int min = x;
if (y < min)
    min = y;
if (z < min)
    min = z;

int max = x;
if (y > max)
    max = y;
if (z > max)
    max = z;

System.out.println("Minimum = " + min);
System.out.println("Maximum = " + max);
System.out.println("Average = " + (x + y + z) / 3.0);
```

2 Program Traces (20 Points Each)

1. For each of the given inputs, write the output of the program.

```
1  import java.util.Scanner;
2
3  public class ExamTrace1_1 {
4
5      public static void main(String[] args) {
6          Scanner keyboard = new Scanner(System.in);
7
8          System.out.print("Input x: ");
9          int x = keyboard.nextInt();
10         System.out.print("Input y: ");
11         int y = keyboard.nextInt();
12         System.out.print("Input z: ");
13         double z = keyboard.nextDouble();
14
15         double w = x / y + z;
16         long t = x % y;
17         int u = ++y - x--;
18         double r = Math.pow(x, 3) + z - y;
19
20         System.out.println(x + "  " + y + "  " + z);
21         System.out.println(w + "  " + t + "  " + u + "  " + r);
22     }
23 }
```

(a)
```
Input x: 2
Input y: 5
Input z: 9
```
Solution:
```
1  6  9.0
9.0  2  4  4.0
```

(b)
```
Input x: 5
Input y: 3
Input z: 6
```
Solution:
```
4  4  6.0
7.0  2  -1  66.0
```

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␣`.

```
1  import java.util.Scanner;
2
3  public class ExamTrace1_2 {
4
5      public static void main(String[] args) {
6          Scanner keyboard = new Scanner(System.in);
7
8          System.out.print("Input string 1: ");
9          String str1 = keyboard.nextLine();
10         System.out.print("Input string 2: ");
11         String str2 = keyboard.nextLine();
12         System.out.print("Input string 3: ");
13         String str3 = keyboard.nextLine();
14
15         int x = str1.indexOf(str2);
16         System.out.println(x);
17         String str4 = str3.substring(x, 2 * x);
18         System.out.println(str4);
19         System.out.println(str1.length());
20
21         int y = str1.indexOf(" ");
22         int z = str1.lastIndexOf(" ");
23         String str6 = str1.substring(0, y);
24         String str5 = str1.substring(z + 1) + str1.substring(y, z) + str6;
25         System.out.println(str5);
26
27         System.out.println(str1.toLowerCase());
28         System.out.println(str1.indexOf('t'));
29         System.out.println(str3.startsWith(str6));
30         System.out.println(str1.startsWith(str6));
31
32         String str7 = str1.replaceAll("t", "R");
33         str7 = str7.replaceAll("s", "Q");
34         str7 = str7.replaceFirst("w", "HELP");
35         System.out.println(str7);
36         System.out.println(str7.substring(str2.length(), str7.length() - str2.length()));
37     }
38 }
```

```
Input string 1: This is a test.
Input string 2: es
Input string 3: Well, I have never been so insulted!
```

Solution:

```
11
e␣never␣bee
15
test.␣is␣aThis
this␣is␣a␣test.
10
false
true
ThiQ␣iQ␣a␣ReQR.
iQ␣iQ␣a␣ReQ
```

3 Coding (20 Points Each)

1. Write a program that will take as input a decimal number representing the user's yearly taxable income and return the amount of income tax they must pay the government. Income tax is calculated as a percentage of the users income according to what bracket they fall in. Here is the tax scheme. If the person makes less than $50,000 they pay 20% of their income in tax. If the person makes $50,000 or more up to but not including $75,000 they pay $10,000 plus 25% of their income that exceeds $50,000. If the person makes $75,000 or more up to but not including $150,000 they pay $16,250 plus 27.5% of their income that exceeds $75,000 in tax. If the person makes $150,000 or more they pay $36,875 plus 30% of their income that exceeds $150,000 in tax. A sample run is below,

```
Input your income: 80000
Your tax is: $17625.00
```

```java
import java.util.Scanner;

public class Test002 {

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

        double tax = 0;

        if (income < 50000) {
            tax = income * 0.2;
        } else if (income < 75000) {
            tax = 10000 + (income - 50000) * 0.25;
        } else if (income < 150000) {
            tax = 16250 + (income - 75000) * 0.275;
        } else {
            tax = 36875 + (income - 150000) * 0.3;
        }

        System.out.printf("Your tax is: $%.2f \n", tax);
    }
}
```

2. Write a program that will solve a quadratic equation given the values of the coefficients. Specifically, the program will solve the equation $ax^2 + bx + c = 0$ for the given values of a, b, and c. Recall from algebra that the solutions to $ax^2 + bx + c = 0$ are given by the quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

   If $a = 0$ then the equation is, $bx + c = 0$ which has a solution of $x = -\frac{c}{b}$. If $b = 0$ as well then you cannot do this calculation. In this case, we have the equation $c = 0$, so if the value of c is 0 then the equation is valid for all possible values of x and if not there are no solutions. If $a \neq 0$ then we could apply the formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$, but if $b^2 - 4ac < 0$ then the square root function in the Math package will not be able to evaluate $\sqrt{b^2 - 4ac}$. If the value of $b^2 - 4ac \geq 0$, if it is then you can apply the quadratic formula and get the two solutions. On the other hand, if $b^2 - 4ac < 0$ then the two solutions are complex numbers, simply tell the user that the solutions are complex.

```java
import java.util.Scanner;

public class LabExample {

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

        System.out.println("Quadratic equation solver for ax^2+bx+c = 0");
        System.out.print("Input a: ");
        double a = kb.nextDouble();
        System.out.print("Input b: ");
        double b = kb.nextDouble();
        System.out.print("Input c: ");
        double c = kb.nextDouble();

        if (a == 0) {
            if (b == 0) {
                if (c == 0)
                    System.out.println("All real numbers are solutions.");
                else
                    System.out.println("No Solution");
            } else
                System.out.println("Solution: x = " + -c / b);
        } else {
            if (b * b - 4 * a * c < 0) {
                System.out.println("Solutions are complex.");
            } else {
                double d = b * b - 4 * a * c;
                double x1 = (-b + Math.sqrt(d)) / (2 * a);
                double x2 = (-b - Math.sqrt(d)) / (2 * a);
                System.out.println("Solutions are x = " + x1 + " and x = " + x2);
            }
        }
    }
}
```