```java
package HW1;

public class Exercise1 {

    public static void main(String[] args) {
        //System.out.println("This is a compile time error")
        System.out.println("This is a run time error: " + 5/0);
        System.out.println("This is true: " + false);
    }
}
```

```
/home/westo
a = 0.0
b = 0.0
c = 0.0
d = 0.0

Process fin
```

```java
package HW1;

public class Exercise2 {

    public static void main(String[] args) {
        double a = 0, b = 0, c = 0, d = 0.0;
        System.out.println("a = " + a + "\nb = " + b + "\nc = " + c + "\nd = " + d);
    }
}
```

```
/home/weston/.jdks/ope
23
229
43981
0
0.3333333333333333
0.3333333333333333
0.33333334
1.23123
2147483647
127
-128
-2147483648
-1
-1
117
-2

Process finished with
```

```java
package HW1;

public class Exercise3 {

    public static void main(String[] args) {
        int a = 2147483647;
        byte b = 127;

        System.out.println(0B10111); // Binary
        System.out.println(0345); // Octal
        System.out.println(0xABCD); // Hex
        System.out.println(1 / 3); // Int division
        System.out.println(1 / 3.0); // Double division
        System.out.println(1.0 / 3); // Double Division
        System.out.println(1.0f / 3.0F); // Float division
        System.out.println(123.123E-2); // Scientific notation
        System.out.println(a); // Not sure what you want me to say here it's a variable.
        System.out.println(b); // Same as above except it's a byte.
        b = (byte) (b + 1); // Byte overflow.
        System.out.println(b); // ^
        a = (int) (a + 1); // Integer overflow
        System.out.println(a); // ^
        b = (byte) ((-b) + 127); // Causes an overflow and goes to -1.
        System.out.println(b); // ^
        a = (int) ((-a) + 2147483647); // Same as last one.
        System.out.println(a); // ^
        a = 2147483647;
        b = 127;
        b = (byte) (b + 1270); // Overflow
        System.out.println(b); // ^
        a = (int) (a + 2147483647); // Overflow
        System.out.println(a); // ^
    }
}
```

```
/home/weston/.jdks/openjdk-17
Option 1: Linear Equations.
Option 2: Shopping.
Option 3: Directions.
Option 4: Roots.
Option 5: Quit.
Enter an option: 1

Enter parameter A: 1
Enter parameter B: 2
Enter parameter C: 3
Enter parameter D: 4
Enter parameter E: 5
Enter parameter F: 6
X: -34 Y: -60
Option 1: Linear Equations.
Option 2: Shopping.
Option 3: Directions.
Option 4: Roots.
Option 5: Quit.
Enter an option: 2

Enter parameter Monitors: 1
Enter parameter Keyboards: 2
Enter parameter Mice: 3
Enter parameter CPU: 4
Enter parameter RAM: 5
Enter parameter SSD: 6
Monitor      1  $     100
Keyboard     2  $     100
Mouse        3  $     105
CPU          4  $    2000
RAM          5  $    2000
SSD          6  $    1200
Subtotal:       $    5455
Tax:            $  392.76
Total:          $ 5847.76
```

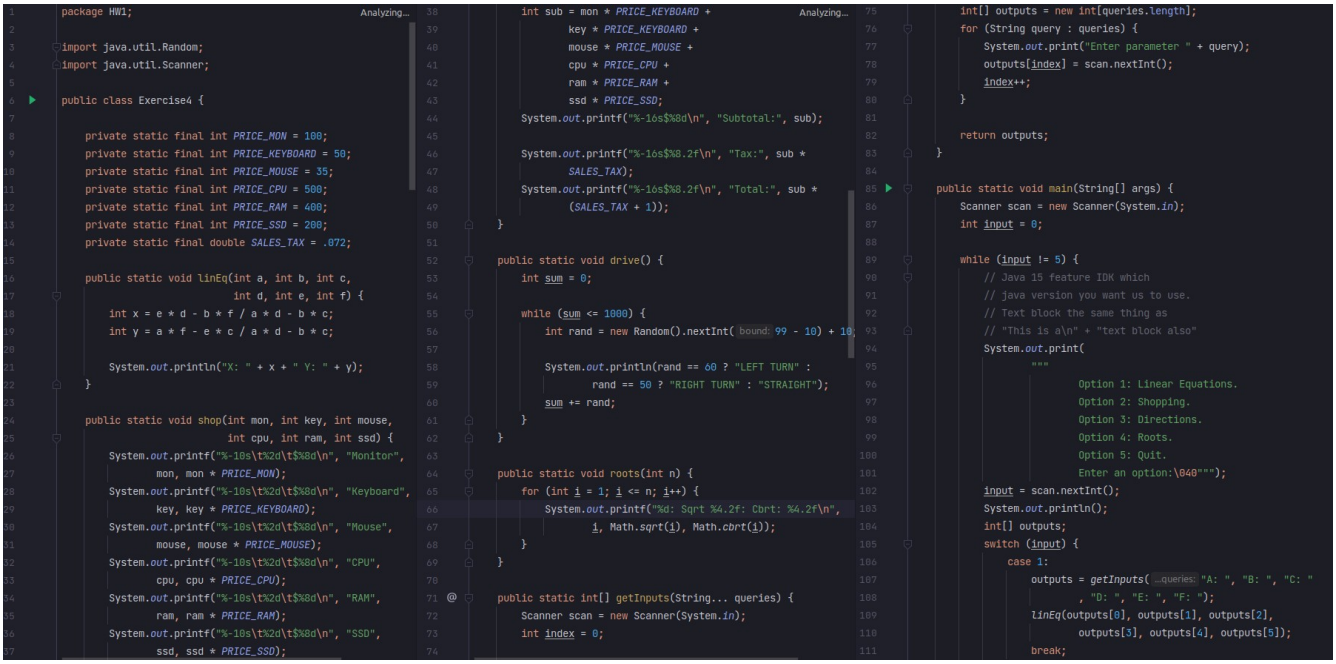
```
Option 1: Linear Equations.
Option 2: Shopping.
Option 3: Directions.
Option 4: Roots.
Option 5: Quit.
Enter an option: 3

STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
STRAIGHT
Option 1: Linear Equations.
Option 2: Shopping.
Option 3: Directions.
Option 4: Roots.
Option 5: Quit.
Enter an option: 4

Enter parameter Enter n: 5
1: Sqrt 1.00: Cbrt: 1.00
2: Sqrt 1.41: Cbrt: 1.26
3: Sqrt 1.73: Cbrt: 1.44
4: Sqrt 2.00: Cbrt: 1.59
5: Sqrt 2.24: Cbrt: 1.71
```

```
Option 1: Linear Equations.
Option 2: Shopping.
Option 3: Directions.
Option 4: Roots.
Option 5: Quit.
Enter an option: 5

Thank you! Exiting.
```



```java
package HW1;

import java.util.Random;
import java.util.Scanner;

public class Exercise4 {

    private static final int PRICE_MON = 100;
    private static final int PRICE_KEYBOARD = 50;
    private static final int PRICE_MOUSE = 35;
    private static final int PRICE_CPU = 500;
    private static final int PRICE_RAM = 400;
    private static final int PRICE_SSD = 200;
    private static final double SALES_TAX = .072;

    public static void linEq(int a, int b, int c,
                             int d, int e, int f) {
        int x = e * d - b * f / a * d - b * c;
        int y = a * f - e * c / a * d - b * c;

        System.out.println("X: " + x + " Y: " + y);
    }

    public static void shop(int mon, int key, int mouse,
                            int cpu, int ram, int ssd) {
        System.out.printf("%-10s\t%2d\t$%8d\n", "Monitor",
                mon, mon * PRICE_MON);
        System.out.printf("%-10s\t%2d\t$%8d\n", "Keyboard",
                key, key * PRICE_KEYBOARD);
        System.out.printf("%-10s\t%2d\t$%8d\n", "Mouse",
                mouse, mouse * PRICE_MOUSE);
        System.out.printf("%-10s\t%2d\t$%8d\n", "CPU",
                cpu, cpu * PRICE_CPU);
        System.out.printf("%-10s\t%2d\t$%8d\n", "RAM",
                ram, ram * PRICE_RAM);
        System.out.printf("%-10s\t%2d\t$%8d\n", "SSD",
                ssd, ssd * PRICE_SSD);
        int sub = mon * PRICE_KEYBOARD +
                key * PRICE_KEYBOARD +
                mouse * PRICE_MOUSE +
                cpu * PRICE_CPU +
                ram * PRICE_RAM +
                ssd * PRICE_SSD;
        System.out.printf("%-10s$%8d\n", "Subtotal:", sub);

        System.out.printf("%-10s$%8.2f\n", "Tax:", sub *
                SALES_TAX);
        System.out.printf("%-10s$%8.2f\n", "Total:", sub *
                (SALES_TAX + 1));
    }

    public static void drive() {
        int sum = 0;

        while (sum <= 1000) {
            int rand = new Random().nextInt( bound: 99 - 10) + 10;

            System.out.println(rand == 60 ? "LEFT TURN" :
                    rand == 50 ? "RIGHT TURN" : "STRAIGHT");
            sum += rand;
        }
    }

    public static void roots(int n) {
        for (int i = 1; i <= n; i++) {
            System.out.printf("%d: Sqrt %4.2f: Cbrt: %4.2f\n",
                    i, Math.sqrt(i), Math.cbrt(i));
        }
    }

    public static int[] getInputs(String... queries) {
        Scanner scan = new Scanner(System.in);
        int index = 0;
        int[] outputs = new int[queries.length];
        for (String query : queries) {
            System.out.print("Enter parameter " + query);
            outputs[index] = scan.nextInt();
            index++;
        }

        return outputs;
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int input = 0;

        while (input != 5) {
            // Java 15 feature IDK which
            // java version you want us to use.
            // Text block the same thing as
            // "This is a\n" + "text block also"
            System.out.print(
                    """
                            Option 1: Linear Equations.
                            Option 2: Shopping.
                            Option 3: Directions.
                            Option 4: Roots.
                            Option 5: Quit.
                            Enter an option:\040""");
            input = scan.nextInt();
            System.out.println();
            int[] outputs;
            switch (input) {
                case 1:
                    outputs = getInputs( ...queries: "A: ", "B: ", "C: "
                            , "D: ", "E: ", "F: ");
                    linEq(outputs[0], outputs[1], outputs[2],
                            outputs[3], outputs[4], outputs[5]);
                    break;
```

```
112                 case 2:
113                     outputs = getInputs( ...queries: "Monitors: ",
114                             "Keyboards: ", "Mice: ", "CPU: ",
115                             "RAM: ", "SSD: ");
116                     shop(outputs[0], outputs[1], outputs[2],
117                             outputs[3], outputs[4], outputs[5]);
118                     break;
119                 case 3:
120                     drive();
121                     break;
122                 case 4:
123                     outputs = getInputs( ...queries: "Enter n: ");
124                     roots(outputs[0]);
125                     break;
126                 case 5:
127                     break;
128                 default:
129                     System.out.println("Incorrect option " +
130                             "please enter another.");
131             }
132         }
133         System.out.println("Thank you! Exiting.");
134     }
135 }
136
```