

```
1 public class JavaTracingMethodsDemo1
2 {
3     public static void main(String[] args)
4     {
5         drawTopFloor();
6
7         for (int i = 0; i < 5; i++)
8         {
9             drawMiddleFloor();
10        }
11
12        drawGroundFloor();
13        drawPeople(5);
14    } // end of main
15
16    public static void drawGroundFloor()
17    {
18        System.out.println("*****");
19        System.out.println("*      *");
20        System.out.println("*   ****   *");
21        System.out.println("* * * * *");
22        System.out.println("* * * * *");
23    } // end of drawGroundFloor
24
25    public static void drawTopFloor() {
26        System.out.println("  **");
27        System.out.println(" *  *");
28        System.out.println(" *  *");
29        System.out.println(" *  *");
30        System.out.println(" *  *");
31    } // end of drawTopFloor
32
33    public static void drawMiddleFloor() {
34        System.out.println("*****"); System.out.println("*      *");
35        System.out.println("*      *"); System.out.println("*      *");
36        System.out.println("*      *");
37    } // end of drawMiddleFloor
38
39    public static void drawPeople(int num)
40    {
41        for (int i = 0; i < num; i++)
42        {
43            System.out.println(" 0");
44            System.out.println(" ---");
45            System.out.println(" |");
46            System.out.println("/ \\");
47            System.out.println();
48        }
49    } // end of drawPeopl
50 }
```

```
1 public class JavaTracingMethodsDemo2
2 {
3     public static void main(String[] args)
4     {
5         System.out.println(10 * 10);
6         System.out.println(computeAreaSquare(4));
7         System.out.println(computePerimeterSquare(5));
8         System.out.println(computeAreaTriangle(5, 8));
9         System.out.println(computeAreaRectangle(10, 4));
10        System.out.println(computeAreaRectangleV2(10, 4));
11        int len = 6;
12        int wid = 8;
13        System.out.println(computeAreaRectangle(len, wid));
14        System.out.println(computeAreaRectangle(len + 1, wid * 2));
15    } // end of main
16
17    public static int computeAreaSquare(int width)
18    {
19        return width * width;
20    } // end of computeAreaSquare
21
22    public static int computePerimeterSquare(int width)
23    {
24        return 4 * width;
25    } // end of computePerimeterSquare
26
27    public static double computeAreaTriangle(int base, int height)
28    {
29        return (base * height) / 2;
30    } // end of computeAreaTriangle
31
32    private static int computeAreaRectangle(int length, int width)
33    {
34        return length * width;
35    } // end of computeAreaRectangle
36
37    public static int computeAreaRectangleV2(int length, int width)
38    {
39        int result = 0;
40        result = length * width;
41        return result;
42    } // end of computeAreaRectangleV2
43 }
```