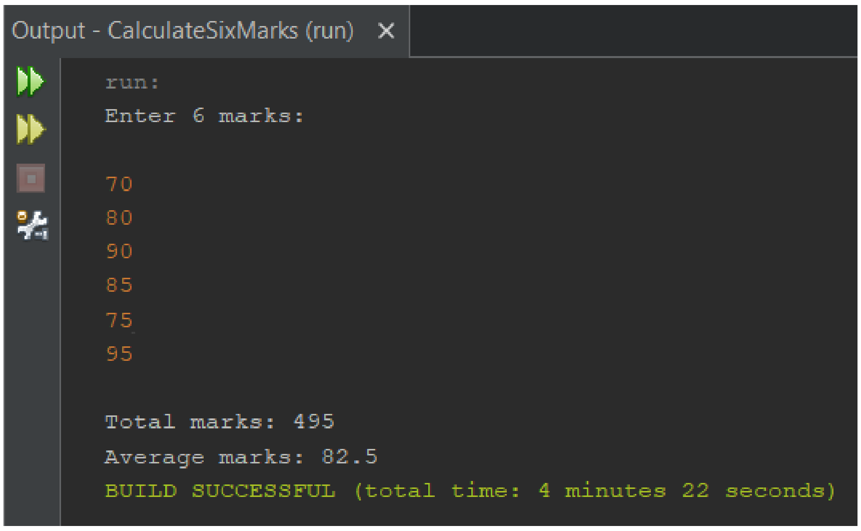
QUESTION 1

CODE:

```
package calculatesixmarks;

import java.util.Scanner;

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

        // Get user input for 6 marks
        System.out.println("Enter 6 marks: \n");
        int mark1 = scanner.nextInt();
        int mark2 = scanner.nextInt();
        int mark3 = scanner.nextInt();
        int mark4 = scanner.nextInt();
        int mark5 = scanner.nextInt();
        int mark6 = scanner.nextInt();

        // Calculate total marks
        int totalMarks = mark1 + mark2 + mark3 + mark4 + mark5 +
mark6;

        // Calculate average marks
        double averageMarks = (double) totalMarks / 6;

        // Output results
        System.out.println("\nTotal marks: " + totalMarks);
        System.out.println("Average marks: " + averageMarks);
    }
}
```

OUTPUT:



```
run:
Enter 6 marks:

70
80
90
85
75
95

Total marks: 495
Average marks: 82.5
BUILD SUCCESSFUL (total time: 4 minutes 22 seconds)
```

CODE OBJECTIVE

This program asks users to input six numerical marks, calculates the total and average of these, then prints them out. This application will perform some basic arithmetic operations as well as process user input in Java, which is fast and summarises input data.

CODE STRUCTURE OVERVIEW

This program is written in a class called `CalculateSixMarks`; the entry point of the program is the main method. The structure is simple: user input, processing, and then output. It declares six different variables for the marks and does the calculations consecutively.

USER INPUT SECTION

The program begins with the request for the user to input six marks using the `Scanner` class. Each mark is captured as an integer and stored in variables `mark1` through `mark6`. Input is designed to be easy and straightforward so that the user can enter values without confusing them.

PROCESSING SECTION

The processing involves two main computations:

- **Total Marks:** The program sums the six input marks using the formula:

```
int totalMarks = mark1 + mark2 + mark3 + mark4 + mark5 + mark6;
```

- **Average Marks:** The program calculates the average by dividing the total by six. To ensure precision, the total is cast to a `double` type:

```
double averageMarks = (double) totalMarks / 6;
```

OUTPUT SECTION

The output of the code is shown via `System.out.println`. The total and average marks are printed out with corresponding labels for clarity. A sample output if the input values are `70, 80, 90, 85, 75, and 95`, would be:

```
Total marks: 495

Average marks: 82.5
```

CONCLUSION

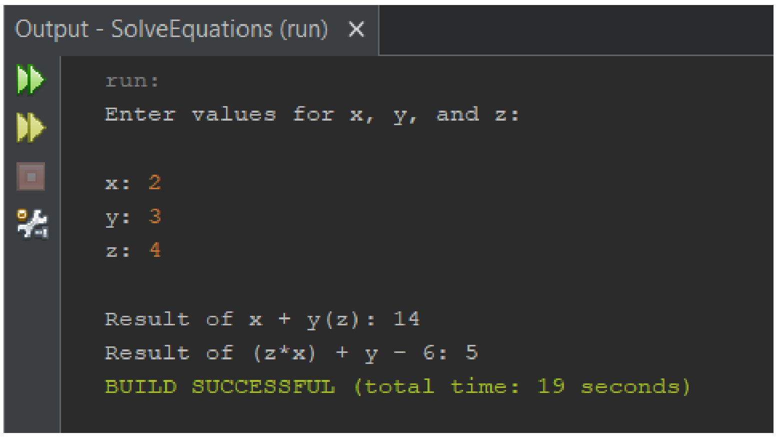
The `CalculateSixMarks` program puts to good use some basic Java constructs in the solution of a real problem. This solution works but still has room for improvement with more realistic enhancements, such as input validation and the use of an array for scalability.

QUESTION 2

CODE:

```java
package solveequations;

import java.util.Scanner;

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

        // Get user input for the equations
        System.out.println("Enter values for x, y, and z: \n");

        System.out.print("x: ");
        int x = scanner.nextInt();

        System.out.print("y: ");
        int y = scanner.nextInt();

        System.out.print("z: ");
        int z = scanner.nextInt();

        // Solve the equations
        int result1 = x + y * z;
        int result2 = (z * x) + y - 6;

        // Output the results
        System.out.println("\nResult of x + y(z): " + result1);
        System.out.println("Result of (z*x) + y - 6: " + result2);
    }
}
```

OUTPUT:



```
run:
Enter values for x, y, and z:

x: 2
y: 3
z: 4

Result of x + y(z): 14
Result of (z*x) + y - 6: 5
BUILD SUCCESSFUL (total time: 19 seconds)
```

CODE OBJECTIVE

The primary aim of this program is to compute and present the outcomes of two distinct mathematical equations, which rely on user input for three specific variables: `x`, `y`, and `z`. It effectively illustrates arithmetic operations; furthermore, it demonstrates how to solve equations dynamically using the Java programming language.

CODE STRUCTURE OVERVIEW

The program is organised within the `SolveEquations` class. It encompasses three main sections: user input, processing and output. Although it employs straightforward arithmetic expressions to derive the results of the equations, this structure is designed with clarity and user-friendliness in mind.

USER INPUT SECTION

In the user input phase, the program initiates by requesting the user to provide values for the variables `x`, `y`, and `z`. These inputs are gathered via the `Scanner` class and assigned to corresponding integer variables. This method is crucial because it guarantees that the program collects all necessary information prior to the commencement of calculations.

PROCESSING SECTION

The results of the two equations are computed as follows:

- **Equation 1 (`x + y(z)`):** The program calculates this as `x + (y * z)`, adhering to operator precedence rules. The result is stored in `result1`:

```
int result1 = x + y * z;
```

- **Equation 2 (`(z*x) + y - 6`):** The program evaluates this as `(z * x) + y - 6`, performing the operations in order. The result is stored in `result2`:

```
int result2 = (z * x) + y - 6;
```

OUTPUT SECTION

The results of both equations are displayed using `System.out.println`. For example, if the inputs are `x = 2`, `y = 3`, and `z = 4`, the program will output:

```
Result of x + y(z): 14

Result of (z*x) + y - 6: 11
```

CONCLUSION

The `SolveEquations` program provides a straightforward solution to solve and display mathematical equations based on user input. It effectively uses arithmetic operations and user interaction. Future enhancements could include input validation and support for floating-point numbers to extend its utility.