Program 1

```
import java.util.Scanner;

public class Performance {
    private int[] marks;

    // Constructor
    public Performance() {
        marks = new int[10];
    }

    // Method to read marks into the array
    public void readMarks() {
        try (Scanner scanner = new Scanner(System.in)) {
            System.out.println("Enter the marks of 10 students:");
            for (int i = 0; i < 10; i++) {
                System.out.print("Student " + (i + 1) + ": ");
                marks[i] = scanner.nextInt();

                // Validate marks are within the given range
                if (marks[i] < 0 || marks[i] > 100) {
                    System.out.println("Invalid mark. Marks should be between
0 and 100. Please enter again.");
                    i--; // Decrement i to re-enter the current student's mark
                }
            }
        }
    }

    // Method to return the highest mark scored in the class
    public int highestmark() {
        int max = marks[0];
        for (int i = 1; i < marks.length; i++) {
            if (marks[i] > max) {
                max = marks[i];
            }
        }
        return max;
    }

    // Method to return the least mark scored in the class
    public int leastmark() {
        int min = marks[0];
        for (int i = 1; i < marks.length; i++) {
            if (marks[i] < min) {
                min = marks[i];
            }
        }
```

```
        return min;
    }

    // Method to return the mode
    public int getMode() {
        int mode = marks[0];
        int maxFrequency = 1;

        for (int i = 0; i < marks.length; i++) {
            int currentMark = marks[i];
            int currentFrequency = 1;

            for (int j = i + 1; j < marks.length; j++) {
                if (marks[j] == currentMark) {
                    currentFrequency++;
                }
            }

            if (currentFrequency > maxFrequency || (currentFrequency ==
maxFrequency && currentMark > mode)) {
                mode = currentMark;
                maxFrequency = currentFrequency;
            }
        }

        return mode;
    }

    // Method to return the frequency at mode
    public int getFreqAtMode() {
        int mode = getMode();
        int frequency = 0;

        for (int i = 0; i < marks.length; i++) {
            if (marks[i] == mode) {
                frequency++;
            }
        }

        return frequency;
    }

    // Method to display the result
    public void display() {
        System.out.println("Highest Mark: " + highestmark());
        System.out.println("Least Mark: " + leastmark());
        System.out.println("Mode: " + getMode());
        System.out.println("Frequency at Mode: " + getFreqAtMode());
```

```
    }

    public static void main(String[] args) {
        Performance performance = new Performance();
        performance.readMarks();
        performance.display();
    }
}
```

Program 2

```
import java.util.Scanner;

public class AlphabetWarGame2 {
    private static final String LEFT_SIDE = "wpbs";
    private static final String RIGHT_SIDE = "mqdz";

    private int getStrength(char letter) {
        if (LEFT_SIDE.indexOf(letter) != -1) {
            switch (letter) {
                case 'w':
                    return 4;
                case 'p':
                    return 3;
                case 'b':
                    return 2;
                case 's':
                    return 1;
            }
        } else if (RIGHT_SIDE.indexOf(letter) != -1) {
            switch (letter) {
                case 'm':
                    return 4;
                case 'q':
                    return 3;
                case 'd':
                    return 2;
                case 'z':
                    return 1;
            }
        }
        return 0; // Default strength for characters not in either side
    }

    public String determineWinner(String word) {
        int leftStrength = 0;
        int rightStrength = 0;
```

```
        for (char letter : word.toCharArray()) {
            int strength = getStrength(letter);
            leftStrength += LEFT_SIDE.indexOf(letter) != -1 ? strength : 0;
            rightStrength += RIGHT_SIDE.indexOf(letter) != -1 ? strength : 0;
        }

        if (leftStrength > rightStrength) {
            return "Left side wins!";
        } else if (rightStrength > leftStrength) {
            return "Right side wins!";
        } else {
            return "Let's fight again!";
        }
    }

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

        System.out.print("Enter a word: ");
        String userInput = scanner.nextLine();

        String result = game.determineWinner(userInput);
        System.out.println(result);
    }
}
```