

Lecture 4

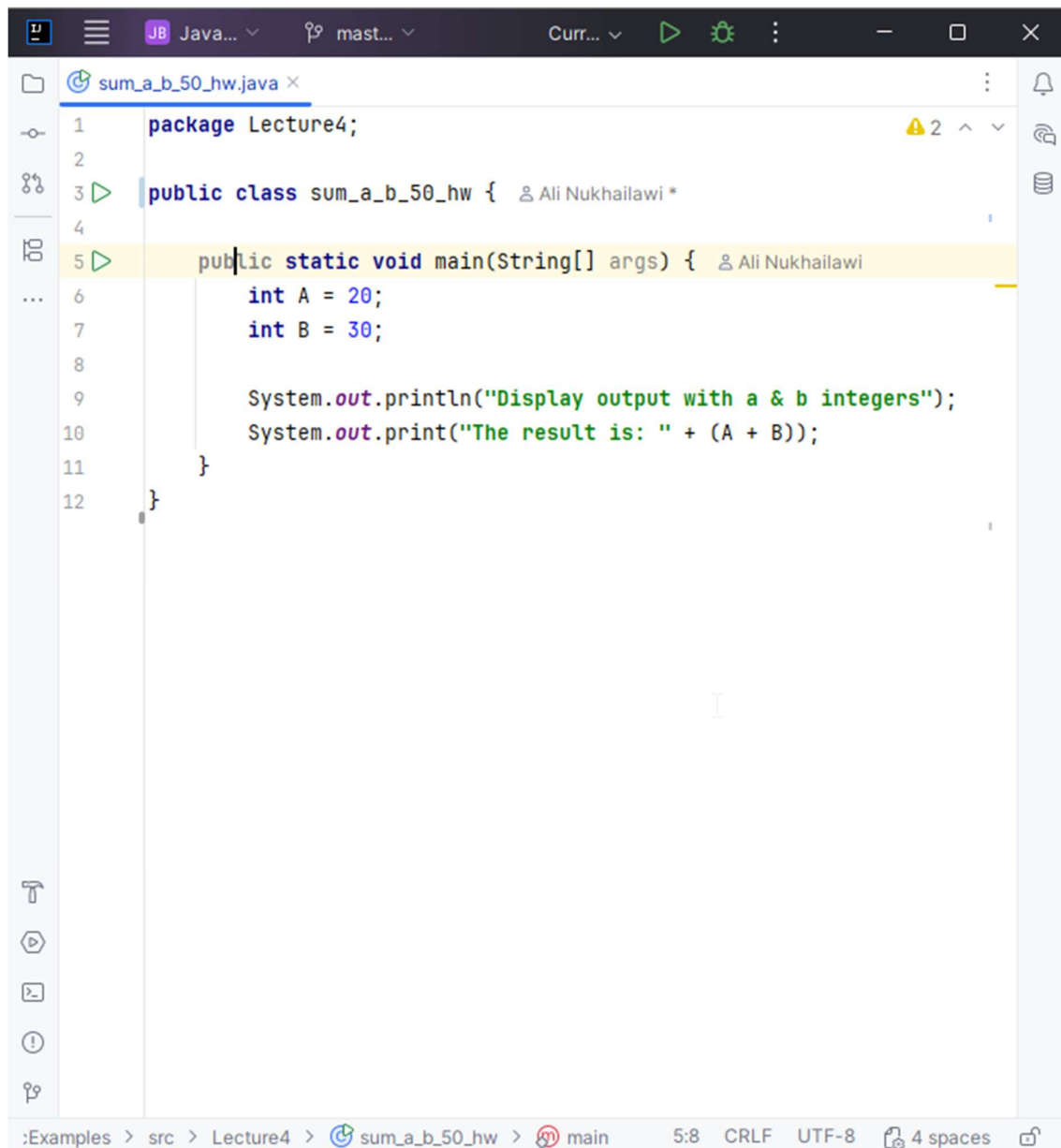
Homework

Use print() with code to get the following output

Int A=20;

Int B=30;

Output: The result is: 50



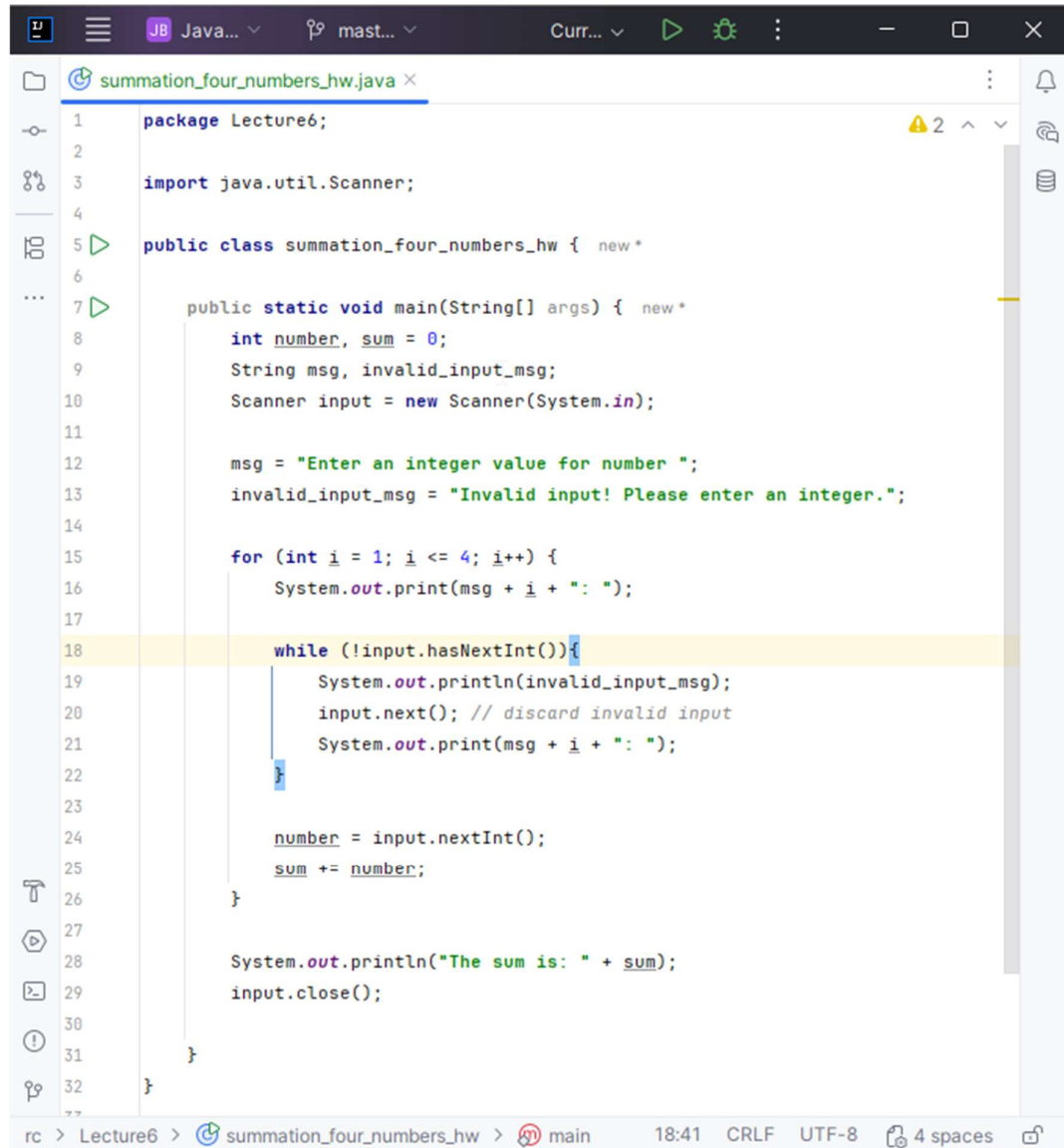
The screenshot shows an IDE window with a Java file named `sum_a_b_50_hw.java`. The code is as follows:

```
1 package Lecture4;
2
3 public class sum_a_b_50_hw {
4
5     public static void main(String[] args) {
6         int A = 20;
7         int B = 30;
8
9         System.out.println("Display output with a & b integers");
10        System.out.print("The result is: " + (A + B));
11    }
12 }
```

The IDE interface includes a top toolbar with icons for file operations, a search bar, and a run button. The left sidebar shows a project tree with the file `sum_a_b_50_hw.java` selected. The bottom status bar displays the file path `:Examples > src > Lecture4 > sum_a_b_50_hw > main`, the line number `5:8`, and encoding settings `CRLF UTF-8 4 spaces`.

Lecture 6

1. Write a program in Java to read four numbers from the user, then find the summation between them?

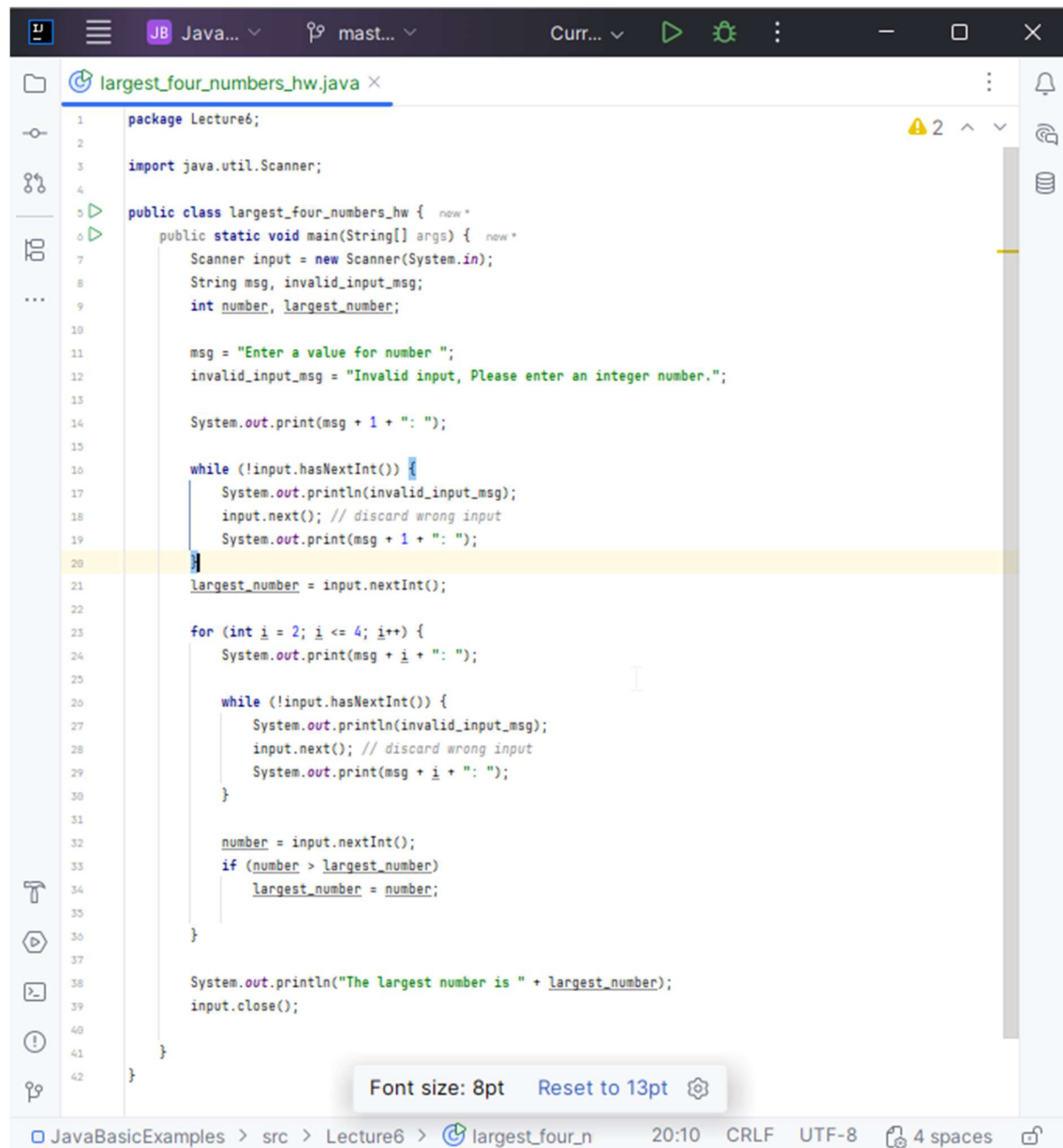


The screenshot shows a Java IDE window titled 'summation_four_numbers_hw.java'. The code is as follows:

```
1 package Lecture6;
2
3 import java.util.Scanner;
4
5 public class summation_four_numbers_hw { new *
6
7     public static void main(String[] args) { new *
8         int number, sum = 0;
9         String msg, invalid_input_msg;
10        Scanner input = new Scanner(System.in);
11
12        msg = "Enter an integer value for number ";
13        invalid_input_msg = "Invalid input! Please enter an integer.";
14
15        for (int i = 1; i <= 4; i++) {
16            System.out.print(msg + i + ": ");
17
18            while (!input.hasNextInt()){
19                System.out.println(invalid_input_msg);
20                input.next(); // discard invalid input
21                System.out.print(msg + i + ": ");
22            }
23
24            number = input.nextInt();
25            sum += number;
26        }
27
28        System.out.println("The sum is: " + sum);
29        input.close();
30    }
31 }
32 }
```

The status bar at the bottom indicates the file path 'rc > Lecture6 > summation_four_numbers_hw > main', the time '18:41', and encoding settings 'CRLF UTF-8 4 spaces'.

2. Write a program in Java to read four numbers from the user, then find the largest number between them?



The screenshot shows an IDE window titled "largest_four_numbers_hw.java". The code is as follows:

```
1 package Lecture6;
2
3 import java.util.Scanner;
4
5 public class largest_four_numbers_hw {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8         String msg, invalid_input_msg;
9         int number, largest_number;
10
11         msg = "Enter a value for number ";
12         invalid_input_msg = "Invalid input, Please enter an integer number.";
13
14         System.out.print(msg + 1 + ": ");
15
16         while (!input.hasNextInt()) {
17             System.out.println(invalid_input_msg);
18             input.next(); // discard wrong input
19             System.out.print(msg + 1 + ": ");
20         }
21         largest_number = input.nextInt();
22
23         for (int i = 2; i <= 4; i++) {
24             System.out.print(msg + i + ": ");
25
26             while (!input.hasNextInt()) {
27                 System.out.println(invalid_input_msg);
28                 input.next(); // discard wrong input
29                 System.out.print(msg + i + ": ");
30             }
31
32             number = input.nextInt();
33             if (number > largest_number)
34                 largest_number = number;
35         }
36
37         System.out.println("The largest number is " + largest_number);
38         input.close();
39     }
40 }
41
42 }
```

At the bottom of the IDE, there is a status bar showing the file path: "JavaBasicExamples > src > Lecture6 > largest_four_n". To the right of the path, it shows "20:10", "CRLF", "UTF-8", and "4 spaces". A font size dropdown menu is also visible, showing "Font size: 8pt" and a "Reset to 13pt" button.

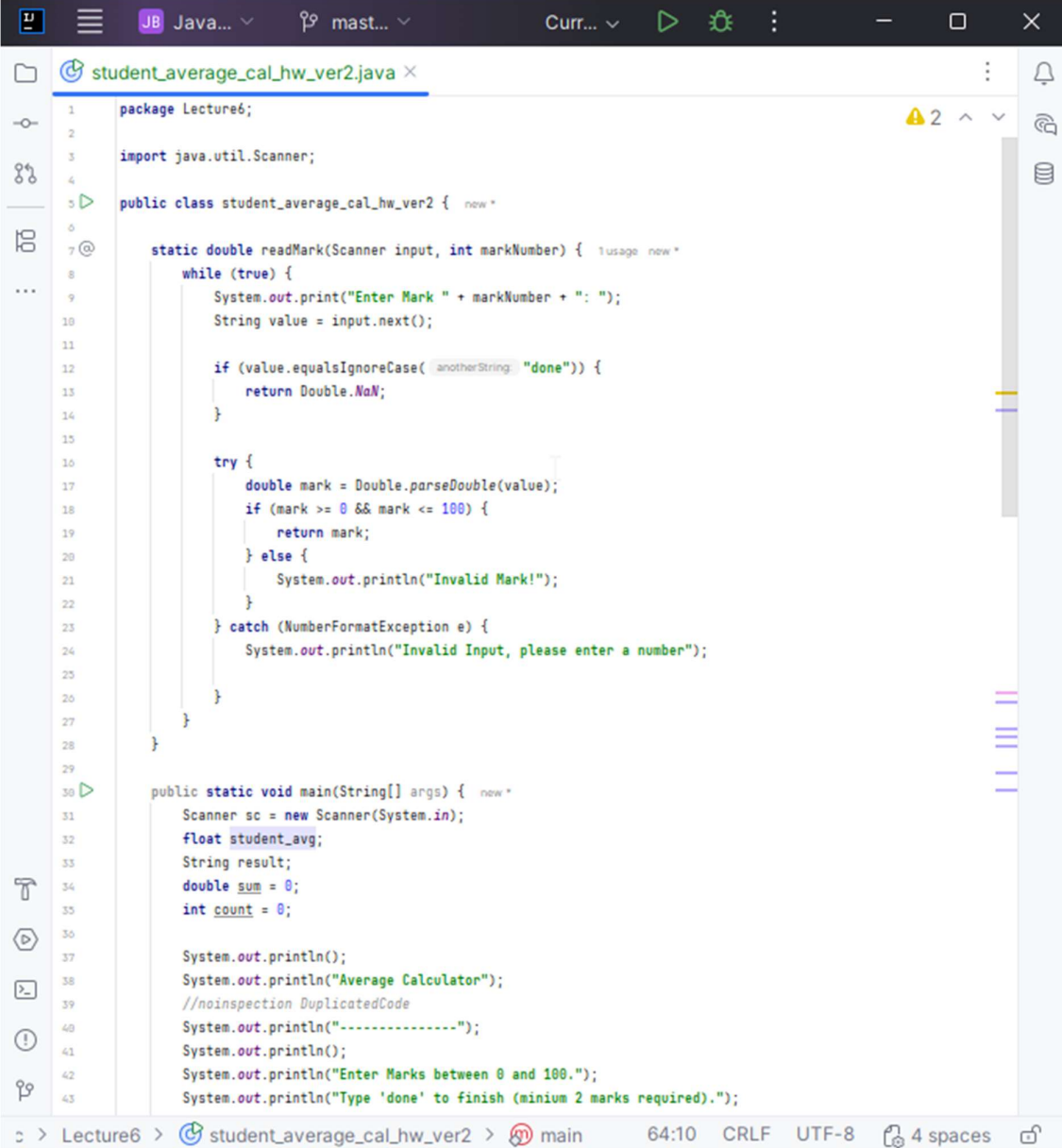
3. Write a program in Java to calculate the value of Y from the following equation?

$$Y = 1/X^2 - 81$$

```
calculate_value_of_y_hw.java x
1 package Lecture6;
2
3 import java.util.Scanner;
4
5 public class calculate_value_of_y_hw { new *
6     public static void main(String[] args) { new *
7         String program_name, input_msg, invalid_input_msg, result_msg;
8         float x, result;
9         Scanner input = new Scanner(System.in);
10        program_name = "Y=1/(X^2-81)";
11        input_msg = "Enter a value for X: ";
12        invalid_input_msg = "Invalid input, Please try again.";
13
14        System.out.println(program_name);
15        System.out.println();
16
17
18        System.out.print(input_msg);
19
20        while (!input.hasNextFloat()) {
21            System.out.println(invalid_input_msg);
22            input.next(); // discard wrong input
23            System.out.print(input_msg);
24        }
25        x = input.nextFloat();
26
27        result = 1 / ((x * x) - 81);
28
29        result_msg = "The Value of Y is: " + result;
30        System.out.println(result_msg);
31
32    }
33 }
```

; > src > Lecture6 > calculate_value_of_y_hw > main 29:1 CRLF UTF-8 4 spaces

4. Write a program to print the student average as follows:
- average >100 and average < 0 print "The average is out of range"
 - average >=0 and average < 50 print "The average is Fail"
 - average >=50 and average <= 59 print "The average is Accepted"
 - average >=60 and average <= 69 print "The average is Medium"
 - average >=70 and average <= 79 print "The average is Good"
 - average >=80 and average <= 89 print "The average is Very Good"
 - average >=90 and average <= 100 print "The average is Excellent"

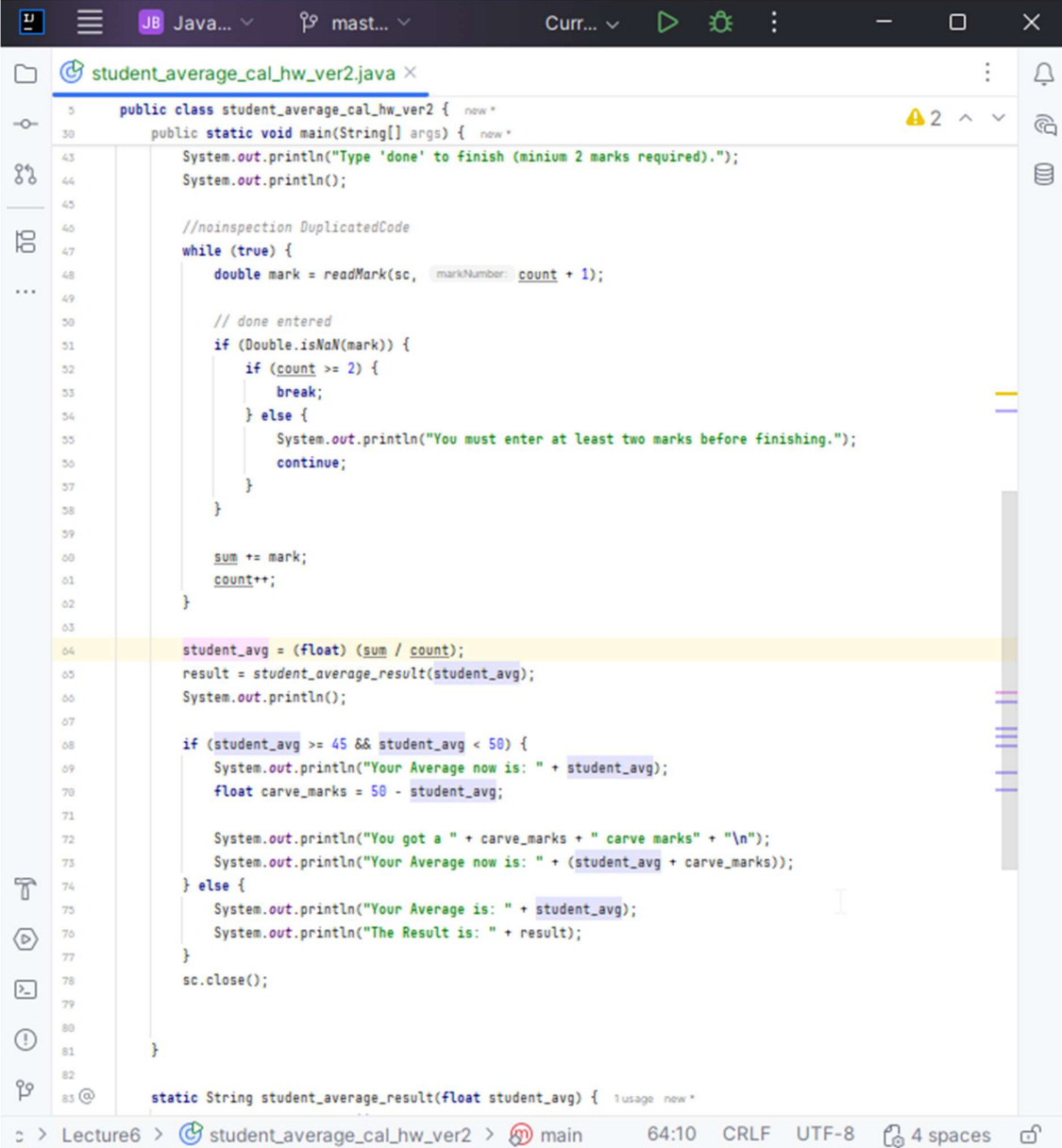


The screenshot shows an IDE window titled "student_average_cal_hw_ver2.java". The code is as follows:

```
1 package Lecture6;
2
3 import java.util.Scanner;
4
5 public class student_average_cal_hw_ver2 {
6
7     static double readMark(Scanner input, int markNumber) {
8         while (true) {
9             System.out.print("Enter Mark " + markNumber + ": ");
10            String value = input.next();
11
12            if (value.equalsIgnoreCase("done")) {
13                return Double.NaN;
14            }
15
16            try {
17                double mark = Double.parseDouble(value);
18                if (mark >= 0 && mark <= 100) {
19                    return mark;
20                } else {
21                    System.out.println("Invalid Mark!");
22                }
23            } catch (NumberFormatException e) {
24                System.out.println("Invalid Input, please enter a number");
25            }
26        }
27    }
28
29
30    public static void main(String[] args) {
31        Scanner sc = new Scanner(System.in);
32        float student_avg;
33        String result;
34        double sum = 0;
35        int count = 0;
36
37        System.out.println();
38        System.out.println("Average Calculator");
39        //noinspection DuplicatedCode
40        System.out.println("-----");
41        System.out.println();
42        System.out.println("Enter Marks between 0 and 100.");
43        System.out.println("Type 'done' to finish (minium 2 marks required).");
```

The IDE interface includes a toolbar at the top with icons for file operations, a search bar, and a status bar at the bottom showing the current file path, line number, and encoding.

(pic 1)



The screenshot shows an IDE window titled "student_average_cal_hw_ver2.java". The code is a Java program that calculates a student's average and result based on marks entered. It includes a while loop for input, a calculation for the average, and conditional logic to determine the result and curved marks. The IDE interface includes a toolbar at the top with icons for file operations, a sidebar on the left with icons for project structure, and a status bar at the bottom showing the current file path and encoding.

```
public class student_average_cal_hw_ver2 {  
    public static void main(String[] args) {  
        System.out.println("Type 'done' to finish (minium 2 marks required).");  
        System.out.println();  
  
        //noinspection DuplicatedCode  
        while (true) {  
            double mark = readMark(sc, markNumber: count + 1);  
  
            // done entered  
            if (Double.isNaN(mark)) {  
                if (count >= 2) {  
                    break;  
                } else {  
                    System.out.println("You must enter at least two marks before finishing.");  
                    continue;  
                }  
            }  
  
            sum += mark;  
            count++;  
        }  
  
        student_avg = (float) (sum / count);  
        result = student_average_result(student_avg);  
        System.out.println();  
  
        if (student_avg >= 45 && student_avg < 50) {  
            System.out.println("Your Average now is: " + student_avg);  
            float carve_marks = 50 - student_avg;  
  
            System.out.println("You got a " + carve_marks + " carve marks" + "\n");  
            System.out.println("Your Average now is: " + (student_avg + carve_marks));  
        } else {  
            System.out.println("Your Average is: " + student_avg);  
            System.out.println("The Result is: " + result);  
        }  
  
        sc.close();  
    }  
  
    static String student_average_result(float student_avg) {
```

(pic 2)

The screenshot shows an IDE window titled "student_average_cal_hw_ver2.java". The code is as follows:

```
5 public class student_average_cal_hw_ver2 {  
30     public static void main(String[] args) {  
71         System.out.println("You got a " + carve_marks + " carve marks" + "\n");  
72         System.out.println("Your Average now is: " + (student_avg + carve_marks));  
73     } else {  
74         System.out.println("Your Average is: " + student_avg);  
75         System.out.println("The Result is: " + result);  
76     }  
77     sc.close();  
78 }  
79  
80  
81  
82  
83 static String student_average_result(float student_avg) {  
84     if (student_avg > 100 || student_avg < 0)  
85         return "Out of Range";  
86     else if (student_avg > 0 && student_avg < 45)  
87         return "Fail";  
88     else if (student_avg > 45 && student_avg < 50)  
89         return "Passed by carve";  
90     else if (student_avg >= 50 && student_avg < 60)  
91         return "Accepted";  
92     else if (student_avg >= 60 && student_avg < 70)  
93         return "Medium";  
94     else if (student_avg >= 70 && student_avg < 80)  
95         return "Good";  
96     else if (student_avg >= 80 && student_avg < 90)  
97         return "Very Good";  
98     else if (student_avg >= 90 && student_avg < 100)  
99         return "Excellent";  
100     else  
101         return "Invalid Input";  
102 }  
103 }  
104  
105  
106 }
```

The IDE interface includes a top toolbar with icons for file operations, a left sidebar with icons for project structure, search, and other tools, and a bottom status bar showing the current file path, line numbers, and encoding settings.

(pic 3)

Lecture 7

1. Write a program to print the student average as follows: using switch case statement:

average >100 and average < 0 print "The average is out of range"

average >=0 and average < 50 print "The average is Fail"

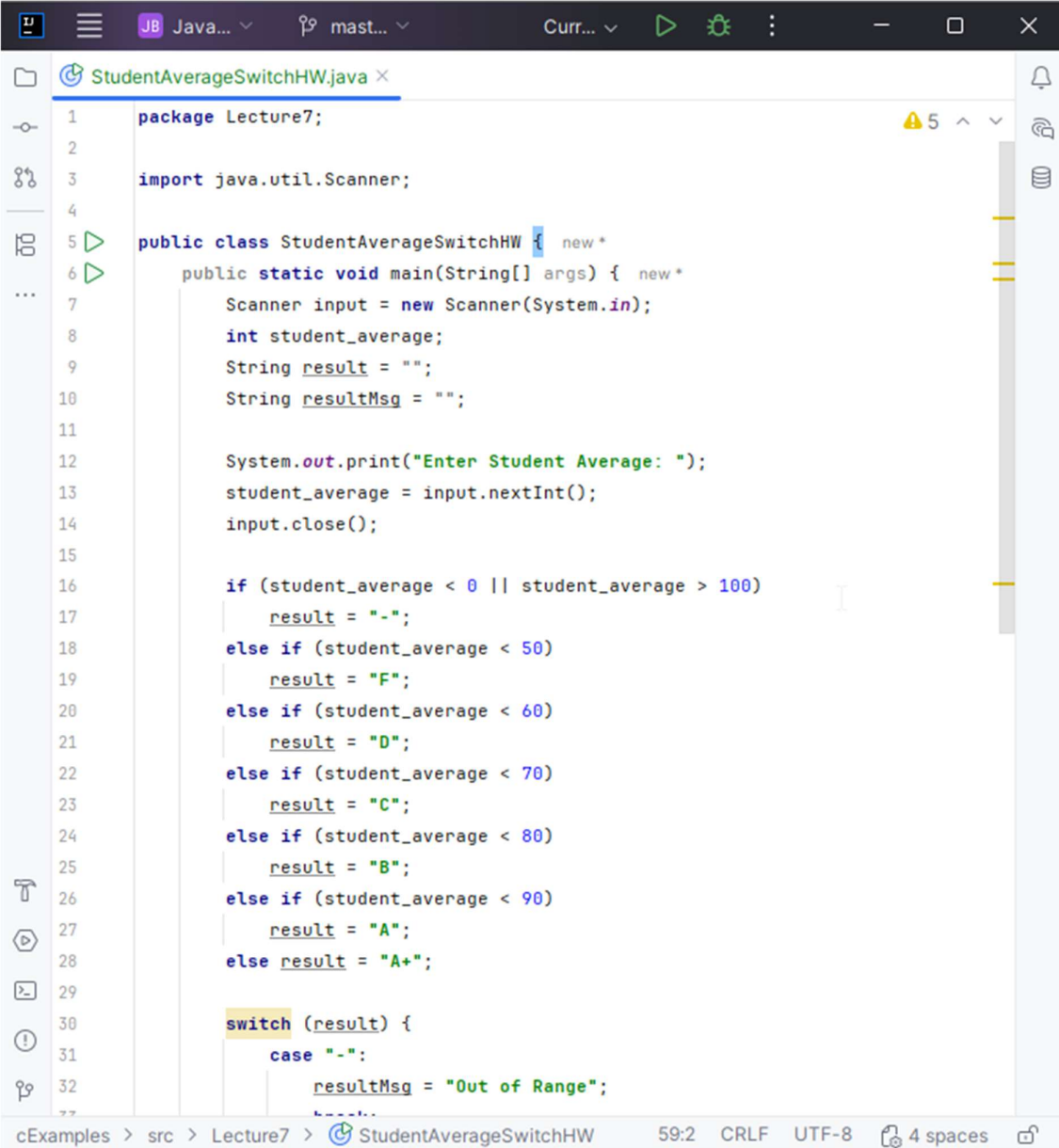
average >=50 and average <= 59 print "The average is Accepted"

average >=60 and average <= 69 print "The average is Medium"

average >=70 and average <= 79 print "The average is Good"

average >=80 and average <= 89 print "The average is Very Good"

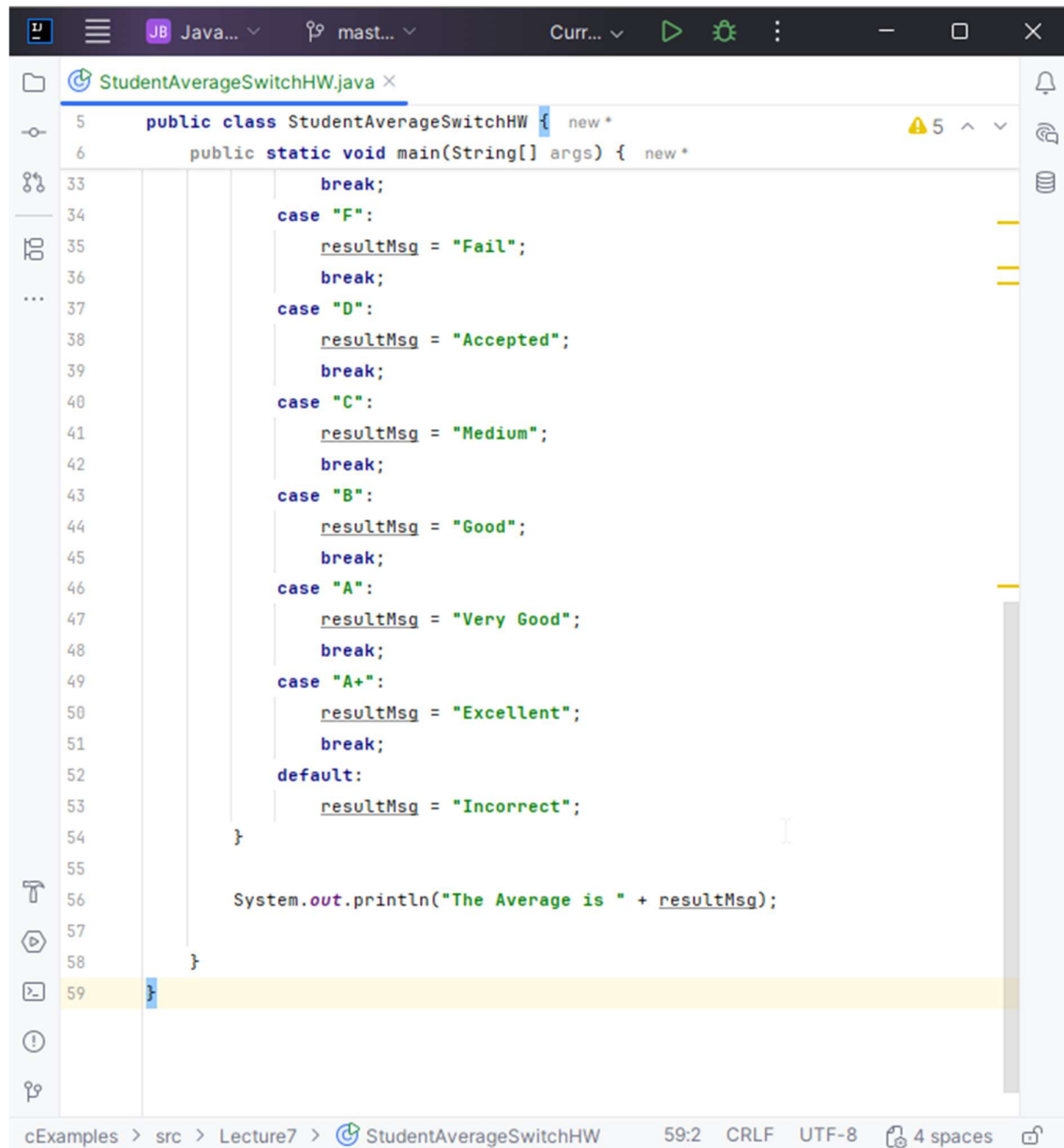
average >=90 and average <= 100 print "The average is Excellent"



```
1 package Lecture7;
2
3 import java.util.Scanner;
4
5 public class StudentAverageSwitchHW { new *
6     public static void main(String[] args) { new *
7         Scanner input = new Scanner(System.in);
8         int student_average;
9         String result = "";
10        String resultMsg = "";
11
12        System.out.print("Enter Student Average: ");
13        student_average = input.nextInt();
14        input.close();
15
16        if (student_average < 0 || student_average > 100)
17            result = "-";
18        else if (student_average < 50)
19            result = "F";
20        else if (student_average < 60)
21            result = "D";
22        else if (student_average < 70)
23            result = "C";
24        else if (student_average < 80)
25            result = "B";
26        else if (student_average < 90)
27            result = "A";
28        else result = "A+";
29
30        switch (result) {
31            case "-":
32                resultMsg = "Out of Range";
33        }
```

The screenshot shows an IDE window titled "StudentAverageSwitchHW.java". The code is a Java program that prompts the user to enter a student average. It uses a series of if-else statements to determine the grade based on the average value. The grades are: Fail (F) for averages between 0 and 50, Accepted (D) for 50-60, Medium (C) for 60-70, Good (B) for 70-80, Very Good (A) for 80-90, and Excellent (A+) for 90-100. A switch statement is used to handle the "Out of Range" case for averages less than 0 or greater than 100. The IDE interface includes a toolbar at the top with icons for file operations, running, and debugging. The bottom status bar shows the current file path, line number (59:2), and encoding (CRLF, UTF-8).

(pic 1)



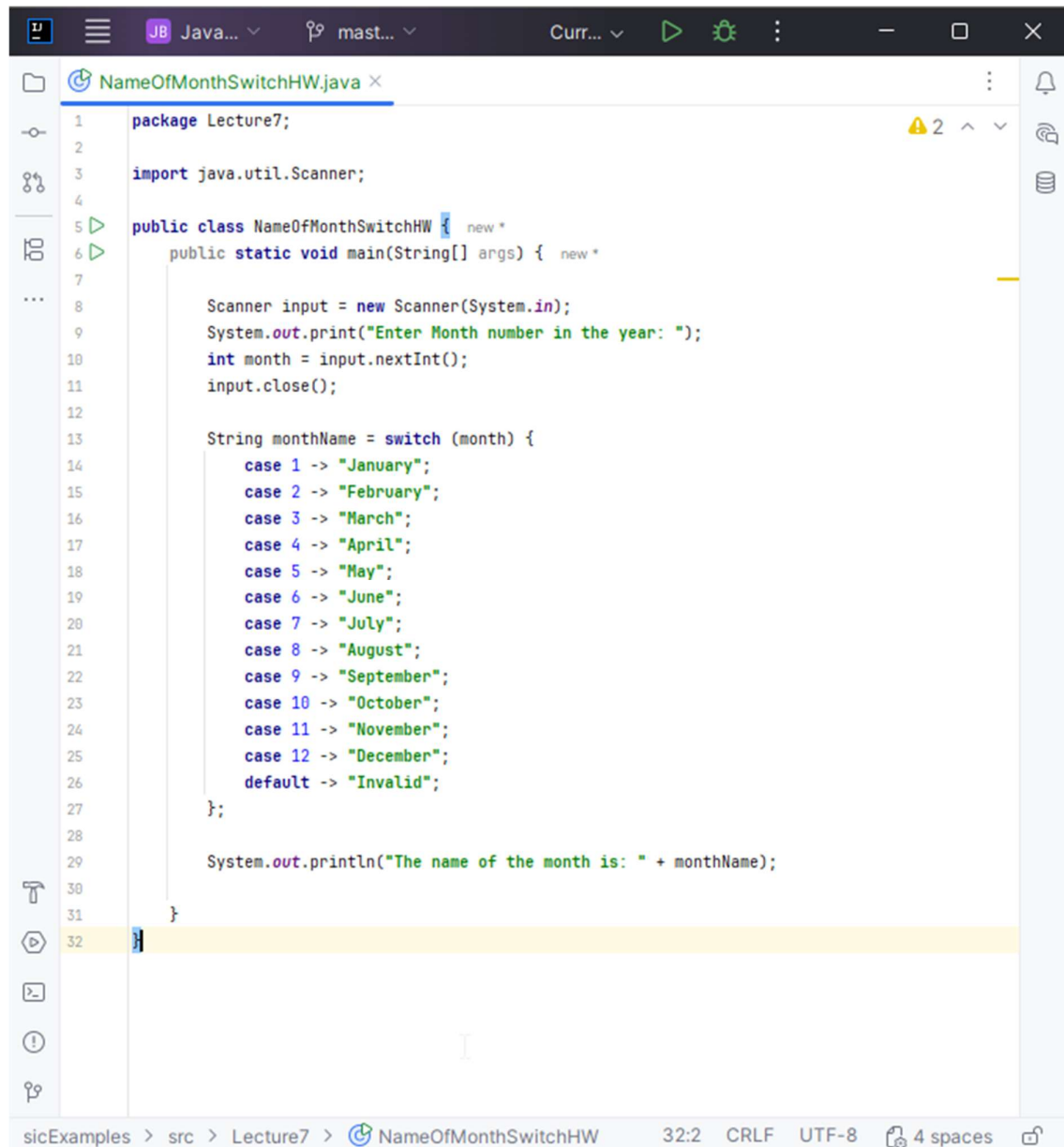
The screenshot shows an IDE window titled "StudentAverageSwitchHW.java". The code is a Java program with a public class and a static main method. Inside the main method, there is a switch statement that maps letter grades to result messages. The grades are F, D, C, B, A, and A+, with corresponding messages "Fail", "Accepted", "Medium", "Good", "Very Good", and "Excellent". A default case returns "Incorrect". After the switch, a println statement outputs the result message. The status bar at the bottom shows the file path "cExamples > src > Lecture7 > StudentAverageSwitchHW", line 59, column 2, and settings for CRLF, UTF-8, and 4 spaces.

```
5 public class StudentAverageSwitchHW { new *
6     public static void main(String[] args) { new *
33         break;
34         case "F":
35             resultMsg = "Fail";
36             break;
37         case "D":
38             resultMsg = "Accepted";
39             break;
40         case "C":
41             resultMsg = "Medium";
42             break;
43         case "B":
44             resultMsg = "Good";
45             break;
46         case "A":
47             resultMsg = "Very Good";
48             break;
49         case "A+":
50             resultMsg = "Excellent";
51             break;
52         default:
53             resultMsg = "Incorrect";
54     }
55
56     System.out.println("The Average is " + resultMsg);
57
58 }
59 }
```

cExamples > src > Lecture7 > StudentAverageSwitchHW 59:2 CRLF UTF-8 4 spaces

(pic 2)

2. Write a program in Java to read number a month of the year, then print the name of the month?



```
1 package Lecture7;
2
3 import java.util.Scanner;
4
5 public class NameOfMonthSwitchHW { new *
6     public static void main(String[] args) { new *
7
8         Scanner input = new Scanner(System.in);
9         System.out.print("Enter Month number in the year: ");
10        int month = input.nextInt();
11        input.close();
12
13        String monthName = switch (month) {
14            case 1 -> "January";
15            case 2 -> "February";
16            case 3 -> "March";
17            case 4 -> "April";
18            case 5 -> "May";
19            case 6 -> "June";
20            case 7 -> "July";
21            case 8 -> "August";
22            case 9 -> "September";
23            case 10 -> "October";
24            case 11 -> "November";
25            case 12 -> "December";
26            default -> "Invalid";
27        };
28
29        System.out.println("The name of the month is: " + monthName);
30    }
31 }
32
```

The screenshot shows a Java IDE with a file named 'NameOfMonthSwitchHW.java'. The code is a Java program that uses a Scanner to read an integer representing a month number. It then uses a switch statement to map the month number to its corresponding name. The names are: 1 for January, 2 for February, 3 for March, 4 for April, 5 for May, 6 for June, 7 for July, 8 for August, 9 for September, 10 for October, 11 for November, 12 for December, and 'Invalid' for any other number. The program prints the name of the month and then returns.