

DEEBAK KUMAR K
192324064
ASSIGNMENT - 6

Question 1 :

Write a Java program to create a method that reads a file and throws an exception if the file is not found

Code :

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class FileReadExample {

    // Method to read a file and print its content to the console
    public static void readFile(String filePath) throws IOException {
        BufferedReader reader = null;
        try {
            reader = new BufferedReader(new FileReader(filePath));
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            throw new IOException("File not found: " + filePath, e);
        } finally {
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    public static void main(String[] args) {
        String filePath = "example.txt"; // Change this to your file path

        try {
            readFile(filePath);
        } catch (IOException e) {
            System.err.println(e.getMessage());
        }
    }
}
```

```

    }
}
}

Main.java
1- import java.io.BufferedReader;
2- import java.io.FileReader;
3- import java.io.IOException;
4
5- public class FileReadExample {
6
7-     // Method to read a file and print its content to the console
8-     public static void readFile(String filePath) throws IOException {
9-         BufferedReader reader = null;
10-        try {
11-            reader = new BufferedReader(new FileReader(filePath));
12-            String line;
13-            while ((line = reader.readLine()) != null) {
14-                System.out.println(line);
15-            }
16-        } catch (IOException e) {
17-            throw new IOException("File not found: " + filePath, e);
18-        } finally {
19-            if (reader != null) {
20-                try {
21-                    reader.close();
22-                } catch (IOException e) {
23-                    e.printStackTrace();
24-                }
25-            }
26-        }
27-    }
28-}

Output
java -cp /tmp/Rf97immGz9/FileReadExample
File not found: example.txt

=== Code Execution Successful ===

```

Question 2 :

Write a Java program to create a class called Student with private instance variables student_id, student_name, and grades. Provide public getter and setter methods to access and modify the student_id and student_name variables. However, provide a method called addGrade() that allows adding a grade to the grades variable while performing additional validation.

Code:

```

import java.util.ArrayList;
import java.util.List;

public class Student {
    private int student_id;
    private String student_name;
    private List<Integer> grades;

    // Constructor
    public Student(int student_id, String student_name) {
        this.student_id = student_id;
        this.student_name = student_name;
        this.grades = new ArrayList<>();
    }
}

```

```

// Getter for student_id
public int getStudentId() {
    return student_id;
}

// Setter for student_id
public void setStudentId(int student_id) {
    this.student_id = student_id;
}

// Getter for student_name
public String getStudentName() {
    return student_name;
}

// Setter for student_name
public void setStudentName(String student_name) {
    this.student_name = student_name;
}

// Method to add a grade with validation
public void addGrade(int grade) {
    if (grade >= 0 && grade <= 100) {
        grades.add(grade);
        System.out.println("Grade added: " + grade);
    } else {
        System.out.println("Invalid grade: " + grade + ". Please enter a grade
between 0 and 100.");
    }
}

// Method to get all grades
public List<Integer> getGrades() {
    return grades;
}

// Main method to test the class
public static void main(String[] args) {
    Student student = new Student(1, "John Doe");

    // Testing getter and setter methods
    System.out.println("Student ID: " + student.getStudentId());
    System.out.println("Student Name: " + student.getStudentName());

    student.setStudentId(2);
    student.setStudentName("Jane Doe");

    System.out.println("Updated Student ID: " + student.getStudentId());
    System.out.println("Updated      Student      Name:      "      +
student.getStudentName());

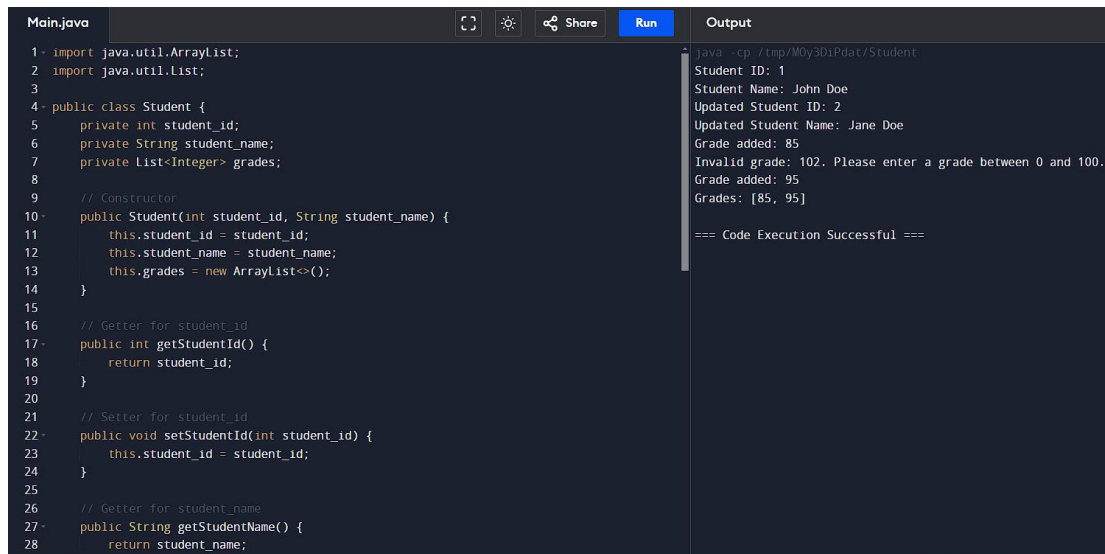
```

```

        // Testing addGrade method
        student.addGrade(85);
        student.addGrade(102); // Invalid grade
        student.addGrade(95);

        // Displaying all grades
        System.out.println("Grades: " + student.getGrades());
    }
}

```



```

Main.java
1- import java.util.ArrayList;
2 import java.util.List;
3
4- public class Student {
5     private int student_id;
6     private String student_name;
7     private List<Integer> grades;
8
9     // Constructor
10-    public Student(int student_id, String student_name) {
11        this.student_id = student_id;
12        this.student_name = student_name;
13        this.grades = new ArrayList<>();
14    }
15
16    // Getter for student_id
17-    public int getStudentId() {
18        return student_id;
19    }
20
21    // Setter for student_id
22-    public void setStudentId(int student_id) {
23        this.student_id = student_id;
24    }
25
26    // Getter for student_name
27-    public String getStudentName() {
28        return student_name;
29    }
30
31-}

```

```

java -cp /tmp/M0y3D1Pdat/Student
Student ID: 1
Student Name: John Doe
Updated Student ID: 2
Updated Student Name: Jane Doe
Grade added: 85
Invalid grade: 102. Please enter a grade between 0 and 100.
Grade added: 95
Grades: [85, 95]

=== Code Execution Successful ===

```

Question 3 :

Write a JavaFX application with a text input field and a button. When the button is clicked, display the text entered in the input field in a label.

Code :

```

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class TextInputDisplayApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        // Create a TextField for user input
        TextField textField = new TextField();

```

```

textField.setPromptText("Enter some text");

// Create a Label to display the text
Label displayLabel = new Label();

// Create a Button to trigger the display action
Button displayButton = new Button("Display Text");

// Set the button action to display the text from the TextField in the Label
displayButton.setOnAction(event -> {
    String text = textField.getText();
    displayLabel.setText(text);
});

// Create a VBox layout and add the TextField, Button, and Label
VBox vbox = new VBox(10, textField, displayButton, displayLabel);

// Create a Scene and set it on the primary stage
Scene scene = new Scene(vbox, 300, 200);
primaryStage.setTitle("Text Input Display Application");
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args) {
    launch(args);
}
}

```