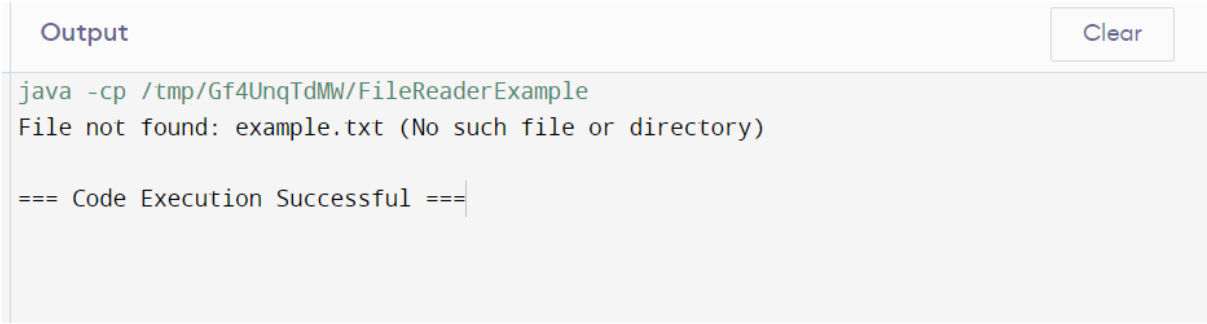
JAVA ASSIGNMENT 6

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.File;

import java.io.FileNotFoundException;

import java.util.Scanner;


public class FileReaderExample {

   public static void main(String[] args) {

      try {

         readFile("example.txt");

      } catch (FileNotFoundException e) {

         System.out.println("File not found: " + e.getMessage());

      }

   }


   public static void readFile(String fileName) throws FileNotFoundException {

      File file = new File(fileName);

      Scanner scanner = new Scanner(file);

      while (scanner.hasNextLine()) {

         System.out.println(scanner.nextLine());

      }

      scanner.close();

   }

}
```

OUTPUT:



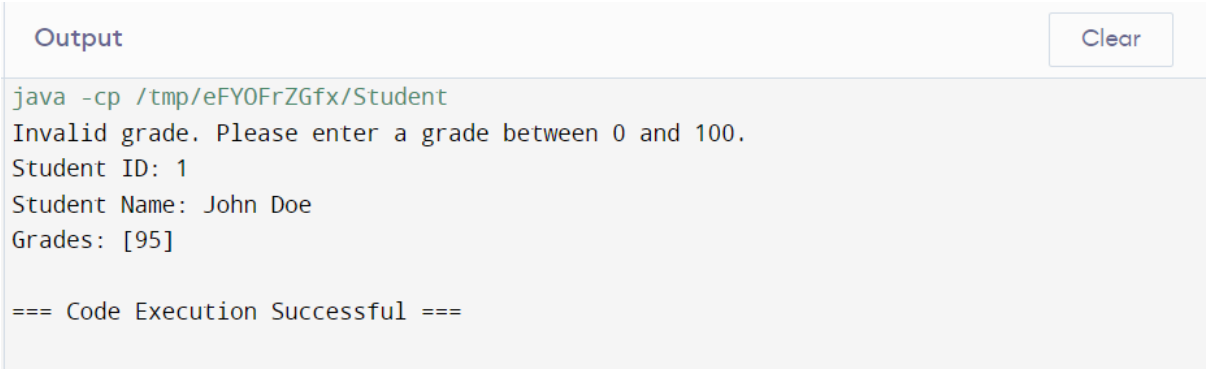
2. Write a Java program to create a method that reads a file and throws an exception if the file is not found

CODE:

```
import java.util.ArrayList;

public class Student {
   private int studentId;
   private String studentName;
   private ArrayList<Integer> grades;

   public Student(int studentId, String studentName) {
      this.studentId = studentId;
      this.studentName = studentName;
      this.grades = new ArrayList<>();
   }

   public int getStudentId() {
      return studentId;
   }

   public void setStudentId(int studentId) {
      this.studentId = studentId;
   }

   public String getStudentName() {
      return studentName;
   }

   public void setStudentName(String studentName) {
      this.studentName = studentName;
   }

   public void addGrade(int grade) {
      if (grade >= 0 && grade <= 100) {
         grades.add(grade);
      } else {
         System.out.println("Invalid grade. Please enter a grade between 0 and 100.");
      }
```

```
    }

    public ArrayList<Integer> getGrades() {
        return grades;
    }

    public static void main(String[] args) {
        Student student = new Student(1, "John Doe");
        student.addGrade(95);
        student.addGrade(105);  // Invalid grade example
        System.out.println("Student ID: " + student.getStudentId());
        System.out.println("Student Name: " + student.getStudentName());
        System.out.println("Grades: " + student.getGrades());
    }
}
```

OUTPUT:



3. 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;


    public Student(int student_id, String student_name) {
```

```
    this.student_id = student_id;
    this.student_name = student_name;
    this.grades = new ArrayList<>();
}

public int getStudentId() {
    return student_id;
}

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

public String getStudentName() {
    return student_name;
}

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

public void addGrade(int grade) {
    if (grade >= 0 && grade <= 100) {
        grades.add(grade);
    } else {
        System.out.println("Invalid grade. Please enter a grade between 0 and 100.");
    }
}

public List<Integer> getGrades() {
    return grades;
```

```
    }

    public static void main(String[] args) {
        Student student = new Student(1, "John Doe");
        student.addGrade(85);
        student.addGrade(90);
        student.addGrade(105);

        System.out.println("Student ID: " + student.getStudentId());
        System.out.println("Student Name: " + student.getStudentName());
        System.out.println("Grades: " + student.getGrades());
    }
}
```

OUTPUT:

Output Clear

```
java -cp /tmp/o58leZycoG/Student
Invalid grade. Please enter a grade between 0 and 100.
Student ID: 1
Student Name: John Doe
Grades: [85, 90]

=== Code Execution Successful ===
```