Practice-5.1

Code:

```
import java.nio.file.FileSystems;
import java.nio.file.Path;
import java.io.File;
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.IOException;

public class CombinedTest {

    // Method to test Path and FileSystem
    public static void testPath() {
        // Create an instance of FileSystem
        java.nio.file.FileSystem fs = FileSystems.getDefault();

        // Create a Path instance
        Path path = fs.getPath("C:/JavaProgramming/employees.txt");

        // Print the constructed Path
        System.out.println("Constructed Path: " + path);
    }

    // Method to test file reading using pre-Java 7 I/O classes
    public static void testFileReading() {
        // Define the file path
        String filePath = "C:/JavaProgramming/employees.txt";

        // Create a File object
        File file = new File(filePath);

        // Declare BufferedReader and FileReader
        BufferedReader bufferedReader = null;
        FileReader fileReader = null;

        try {
            // Instantiate FileReader and BufferedReader
            fileReader = new FileReader(file);
            bufferedReader = new BufferedReader(fileReader);

            String line;
```

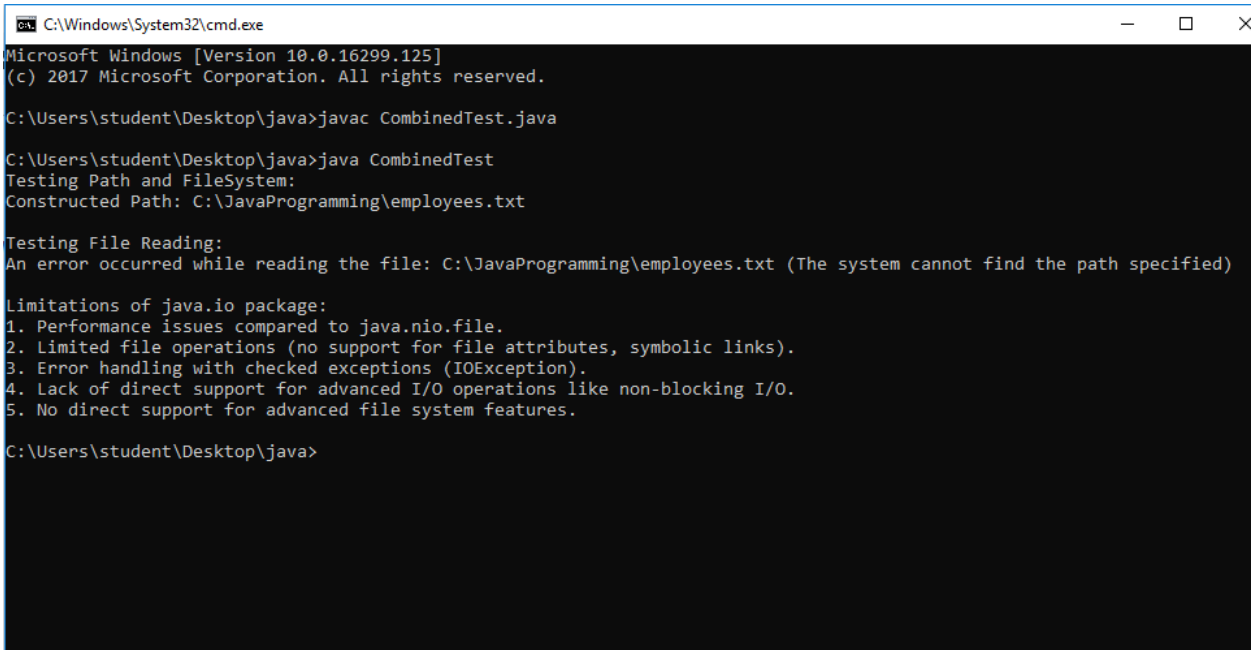
```
            // Read lines from the file
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            // Handle file not found or IO exceptions
            System.err.println("An error occurred while reading the file: " + e.getMessage());
        } finally {
            // Close resources in the finally block to ensure they are closed even if an exception
occurs
            try {
                if (bufferedReader != null) {
                    bufferedReader.close();
                }
                if (fileReader != null) {
                    fileReader.close();
                }
            } catch (IOException ex) {
                System.err.println("An error occurred while closing the resources: " +
ex.getMessage());
            }
        }
    }

    public static void main(String[] args) {
        // Test Path and FileSystem
        System.out.println("Testing Path and FileSystem:");
        testPath();

        // Test file reading
        System.out.println("\nTesting File Reading:");
        testFileReading();

        // Main limitations of the java.io package (commented for reference)
        System.out.println("\nLimitations of java.io package:");
        System.out.println("1. Performance issues compared to java.nio.file.");
        System.out.println("2. Limited file operations (no support for file attributes, symbolic
links).");
        System.out.println("3. Error handling with checked exceptions (IOException).");
        System.out.println("4. Lack of direct support for advanced I/O operations like non-blocking
I/O.");
        System.out.println("5. No direct support for advanced file system features.");
    }
}
```

Output:


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\student\Desktop\java>javac CombinedTest.java
C:\Users\student\Desktop\java>java CombinedTest
Testing Path and FileSystem:
Constructed Path: C:\JavaProgramming\employees.txt
Testing File Reading:
An error occurred while reading the file: C:\JavaProgramming\employees.txt (The system cannot find the path specified)
Limitations of java.io package:
1. Performance issues compared to java.nio.file.
2. Limited file operations (no support for file attributes, symbolic links).
3. Error handling with checked exceptions (IOException).
4. Lack of direct support for advanced I/O operations like non-blocking I/O.
5. No direct support for advanced file system features.
C:\Users\student\Desktop\java>