- 1. Write a program to create a new text file named test.txt.
- 2. Write a program to check whether a file exists at a given path.
- 3. Write a Java program to write "Hello, World!" into a file using FileWriter.
- 4. Write a program to read the content of a file line by line using BufferedReader.
- 5. Write a program to append a line of text to an existing file.
- 6. Write a program to count the number of lines, words, and characters in a file.
- 7. Write a program to copy content from one file to another using FileReader and FileWriter.

```
package Assignment_Day9;

import java.io.*;
import java.util.*;

public class Problem11 {
public static void main(String[] args) {
String fileName = "student1234.txt";

try {
FileWriter writer = new FileWriter(fileName);
writer.write("Hari\n");
writer.write("Kmar\n");
writer.write("eswar\n");
writer.write("vinay\n");
writer.write("rao\n");
writer.close();
System.out.println("file created ");

System.out.println("Reading from file " + fileName);
BufferedReader reader = new BufferedReader(new FileReader(fileName));
String line;
while ((line = reader.readLine()) != null) {
System.out.println(line);
}
reader.close();

writer = new FileWriter(fileName, true);
writer.write("krishna\n");
writer.close();
System.out.println("Data appended Successfully");



int lineCount = 0;
int wordCount = 0;
reader = new BufferedReader(new FileReader(fileName));
while ((line = reader.readLine()) != null) {
```

```
lineCount++;
String[] words = line.split("\\s+");
wordCount += words.length;
}
reader.close();
System.out.println("Number of lines: " + lineCount);
System.out.println("Number of words: " + wordCount);

} catch (Exception e) {
System.out.println(e);
}
}
}
```

```
file created
Reading from file  student1234.txt
Hari
Kmar
eswar
vinay
rao
Data appended Successfully
Number of lines: 6
Number of words: 6
```

Q5. Copy Contents from One File to Another

Write a program to read from source.txt and write the same content into destination.txt.

Q12. Delete a File

Write a program to delete a file (given by file name) if it exists.

```
package File_Handling;


import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;

public class CopyFile {

public static void main(String[] args) throws IOException {
Path path=Paths.get("Sample121.txt");
Files.createFile(path);
System.out.println("File Created");

//copy file
Path copypath=Paths.get("Samplecopy1.txt");
Files.copy(path, copypath,StandardCopyOption.REPLACE_EXISTING);
System.out.println("File Copied");

//Move File
Path movedPath=Paths.get("movedfile1.txt");
Files.move(copypath, movedPath, StandardCopyOption.REPLACE_EXISTING);
System.out.println("File moved");

//Delete File
Files.deleteIfExists(movedPath);
Files.deleteIfExists(path);
System.out.println("File deleted");

}

}
```

```
File Created
File Copied
File moved
File deleted
```

Q6. Check if a File Exists and Display Properties

Create a program to check if report.txt exists. If it does, display its:

- Absolute path
- File name
- Writable (true/false)
- Readable (true/false)
- File size in bytes

```
package File_Handling;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;

public class all_pract {

    public static void main(String[] args) throws IOException {
```

```
    File f=new File("sample1.txt");
    System.out.println(f.getName());
    System.out.println(f.length());
    System.out.println(f.getAbsolutePath());
    System.out.println(f.canRead());
    System.out.println(f.canWrite());

try {
  BufferedWriter w = new BufferedWriter(new FileWriter("sample1.txt"));
  w.write("Hello");
  w.newLine();
  w.write("Welcome to Java Learning Sessions");
  w.close();
  System.out.println("\nWriting Completed");
} catch (Exception e) {
  System.out.println(e);
}


try {
  List<String> l = new ArrayList<String>();
  BufferedReader r = new BufferedReader(new FileReader("sample1.txt"));
  String line;
  System.out.println("\nReading File");
  while ((line = r.readLine()) != null) {
    l.add(line);
  }
  for (String k : l) {
    System.out.println(k);
  }
  r.close();
} catch (Exception e) {
  System.out.println(e);
}


try {
  PrintWriter p = new PrintWriter(new FileWriter("sample1.txt"));
```

```
            p.println("Hello world");
            p.print("Using PrintWriter");
            p.close();
            System.out.println("\nWriting Completed");
        } catch (Exception e) {
            System.out.println(e);
        }


        try {
            List<String> l = new ArrayList<String>();
            BufferedReader r = new BufferedReader(new FileReader("sample1.txt"));
            String line;
            System.out.println("\nReading file ");
            while ((line = r.readLine()) != null) {
                l.add(line);
            }
            for (String k : l) {
                System.out.println(k);
            }
            r.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
sample1.txt
30
C:\Users\user\Desktop\java1807\java_practice\sample1.txt
true
true

Writing Completed

Reading File
Hello
Welcome to Java Learning Sessions

Writing Completed

Reading file
Hello world
Using PrintWriter
```

- 8. Write a program that lists all the files in a directory.
- 9. Write a program to filter and display only .txt files from a folder using FilenameFilter.

```
package Assignment_Day11;

import java.io.File;
import java.io.FilenameFilter;

public class Problem1 {
public static void main(String[] args) {
String directoryPath = ".";
try {
File directory = new File(directoryPath);
System.out.println("All files in directory:");
File[] allFiles = directory.listFiles();
if (allFiles != null) {
for (File file : allFiles) {
if (file.isFile()) {
System.out.println(file.getName());
```

```
}
}
}
System.out.println("-------------------------------------");
System.out.println(".txt files ");
FilenameFilter txtFilter = new FilenameFilter() {
@Override
public boolean accept(File dir, String name) {
return name.toLowerCase().endsWith(".txt");
}
};
File[] txtFiles = directory.listFiles(txtFilter);
if (txtFiles != null) {
for (File file : txtFiles) {
System.out.println(file.getName());
}
}
} catch (Exception e) {
System.out.println( e);
}
}
}
```

```
All files in directory:
.classpath
.gitignore
.project
DataFile.txt
DataFile1.txt
DataFile2.txt
employee
employee.ser
employee.txt
Sample.txt
sample1.txt
Sample12.txt
Samplecopy.txt
Student.txt.txt
student1234.txt
-----------------------------------
.txt files
DataFile.txt
DataFile1.txt
DataFile2.txt
employee.txt
Sample.txt
sample1.txt
Sample12.txt
Samplecopy.txt
Student.txt.txt
student1234.txt
```

10. Write a program to serialize and deserialize a Student object to and from a file.

```
package Assignment_Day11;

import java.io.*;

class Student implements Serializable {
private static final long serialVersionUID = 1L;
String name;
int rollNo;
```

```
public Student(String name, int rollNo) {
this.name = name;
this.rollNo = rollNo;
}

@Override
public String toString() {
return "Student{name='" + name + "', rollNo=" + rollNo + "}";
}
}

public class Problem2 {
public static void main(String[] args) {
Student student = new Student("Hari", 101);
String filename = "student.ser";

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filename)))
{
oos.writeObject(student);
System.out.println("Student object serialized");
} catch (IOException e) {
System.out.println("Error during serialization: " + e.getMessage());
}

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filename))) {
Student deserializedStudent = (Student) ois.readObject();
System.out.println("Deserialized Student: " + deserializedStudent);
} catch (Exception e) {
System.out.println(e);
}
}
}
```

12. Write a program to search for a specific word in a file and count its occurrences.

```
package Assignment_Day11;

import java.io.File;
import java.io.IOException;
import java.util.Scanner;

public class Problem3{
public static void main(String[] args) {
String searchWord = "Hari";
int count = 0;
try (Scanner scanner = new Scanner(new File("student1234.txt"))) {
while (scanner.hasNext()) {
```

```
if (scanner.next().equalsIgnoreCase(searchWord)) {
count++;
}
}
System.out.println("count: " + count);
} catch (Exception e) {
System.out.println(e);
}
}
}
```

```
count: 1
```

15. Write a program to write data into a file using Files.write() and append using StandardOpenOption.APPEND.

```
package Assignment_Day11;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;

public class Problem4 {
public static void main(String[] args) {
try {
Files.write(Paths.get("data.txt"), "Initial content\n".getBytes());
System.out.println("Initial content written");

Files.write(Paths.get("data.txt"), "Appended content\n".getBytes(),
StandardOpenOption.APPEND);
System.out.println("Content appended");
} catch (IOException e) {
System.out.println("Error: " + e.getMessage());
}
}
}
```

```
Initial content written
Content appended
```

19. Write a program to serialize a class Employee and store it in employee.ser.

```
package File_Handling;

import java.io.BufferedWriter;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.ObjectOutputStream;



public class Serial_data {

public static void main(String[] args) {
Employee emp=new Employee(101,"Minisha Shah");
Employee emp2=new Employee(102,"Shah");
//		try

//		{
//			FileWriter fw=new FileWriter("employee.");
//			ObjectOutputStream oos=new ObjectOutputStream(fos);
//			oos.writeObject(emp);
//			oos.close();
//			fos.close();
//			System.out.println("Done");
//		}
try
```

```
{
FileWriter fw=new FileWriter("employee.txt");
BufferedWriter bf=new BufferedWriter(fw);
bf.write("ID\tName\n");
bf.write(emp.toString());
bf.newLine();
bf.write(emp2.toString());
bf.newLine();
bf.close();
fw.close();
System.out.println("Done");
}
catch(Exception e)
{
System.out.println(e);
}
}

}
```

20. Write a program to deserialize the employee.ser file and display the object data.

```
package File_Handling;

import java.io.FileInputStream;
import java.io.ObjectInputStream;
import java.io.FileNotFoundException;
```

```
import java.io.FileOutputStream;
import java.io.IOException;



public class Se_data {


public static void main(String[] args) throws IOException ,ClassNotFoundException{
// TODO Auto-generated method stub
FileInputStream fos=new FileInputStream("employee.txt");
ObjectInputStream oos=new ObjectInputStream(fos);
Student s1=(Student)oos.readObject();
s1.display();
oos.close();
fos.close();
System.out.println("Deserialization done successfully");


}


}
```