Java I/O File Handling -

- **1. Write a program to create a new text file named test.txt.**
  Program:

```
import java.io.*;
public class Main
{
    public static void main(String[] args) throws Exception
    {
            Files.createFile(Paths.get("test.txt"));
            System.out.println("File created successfully!");
    }
}
```

  OutPut:
  File created Successfully

- **2. Write a program to check whether a file exists at a given path.**
  Program:

```
import java.io.File;
public class Main
{
    public static void main(String[] args)
    {
            File f = new File("test.txt"); // file name
            if (f.exists())
            {
                    System.out.println("File exists");
            }
            else
            {
                    System.out.println("File does not exist");
            }
    }
}
```

  OutPut:
  File exists

- **3. Write a Java program to write "Hello, World!" into a file using FileWriter.**
  Program:

```
import java.io.*;

public class Main
{
    public static void main(String[] args) throws IOException
    {
            FileWriter fw = new FileWriter("test.txt");
            fw.write("Hello, World!");
            fw.close();
            System.out.println("Done");
    }
}
```

  OutPut:

```
Done
```

- 4**. Write a program to read the content of a file line by line using BufferedReader.**
  Program:

```
import java.io.*;
public class Main
{
    public static void main(String[] args) throws IOException
    {
            BufferedReader br = new BufferedReader(new FileReader("test.txt"));
            String line;
            while ((line = br.readLine()) != null)
            {
                    System.out.println(line);
            }
            br.close();
    }
}
```

- **5. Write a program to append a line of text to an existing file.**
  Program:

```
import java.io.*;
public class Main
{
    public static void main(String[] args) throws IOException
    {
            FileWriter fw = new FileWriter("test.txt", true);
            fw.write("This is a new line.\n");
            fw.close();
            System.out.println("Text appended successfully.");
    }
}
```

- **6. Write a program to count the number of lines, words, and characters in a file.**

Program:

```
import java.io.*;
public class Main
{
    public static void main(String[] args) throws Exception
    {
            BufferedReader br = new BufferedReader(new FileReader("test.txt"));
            int lines = 0, words = 0, chars = 0;
            String s;
            while ((s = br.readLine()) != null)
            {
                    lines++;
                    words += s.split(" ").length;
                    chars += s.length();
            }
            br.close();
            System.out.println("Lines: " + lines);
            System.out.println("Words: " + words);
            System.out.println("Characters: " + chars);
    }
}
```

- **7. Write a program to copy content from one file to another using FileReader and FileWriter.**

Program:

```
import java.io.*;
public class CopyFile
```

```
{
    public static void main(String[] args) throws Exception
    {
            FileReader fr = new FileReader("source.txt");
            FileWriter fw = new FileWriter("destination.txt");
            int ch;
            while ((ch = fr.read()) != -1)
            {
                    fw.write(ch);
            }
            fr.close();
            fw.close();
            System.out.println("File copied successfully!");
    }
}
```

Output:

File copied successfully!

- **8. Write a program that lists all the files in a directory.**

Program

```
import java.io.File;
public class ListFiles
{
    public static void main(String[] args)
    {
            File folder = new File("C:\\Users\\jayan\\Desktop\\File Handling in java");
            File[] files = folder.listFiles();
            if (files != null)
            {
                            for (File f : files)
                    {
                            System.out.println(f.getName());
                            }
            }
            else
            {
                            System.out.println("Directory not found!");
            }
    }
```

```
}
```
OutPut:

Text.txt
Sample.txt
Employee.txt

- **9. Write a program to filter and display only .txt files from a folder using FilenameFilter.**

Program:

```
import java.io.*;
public class TxtFilter
{
    public static void main(String[] args)
    {
            File folder = new File(".");
            String[] files = folder.list((dir, name) -> name.endsWith(".txt"));
            for (String f : files)
            {
                    System.out.println(f);
            }
    }
}
```
OutPut:

Text.txt
Sample.txt
Employee.txt
Student.txt

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

Program:

```
import java.io.*;
class Student implements Serializable
{
    String name;
    int age;
    Student(String name, int age)
    {
```

```
        this.name = name;
        this.age = age;
    }
}
public class SerializeExample
{
    public static void main(String[] args)
    {
    Student s1 = new Student("Jayanth", 22);
    try
    (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream("student.ser")))
     {
            out.writeObject(s1);
            System.out.println("Object serialized successfully!");
    }
    catch (IOException e)
    {
            e.printStackTrace();
    }
    try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream("student.ser")))
    {
            Student s2 = (Student) in.readObject();
            System.out.println("Object deserialized:");
            System.out.println("Name: " + s2.name + ", Age: " + s2.age);
    }
    catch (IOException | ClassNotFoundException e)
    {
            e.printStackTrace();
    }
  }
}
```

Output:

```
    Object serialized successfully!
    Object deserialized:
    Name: Jayanth, Age: 22
```

- **11. Write a program to read a file using Scanner and display the tokens.**

Program:

```
import java.io.*;
import java.util.*;
public class ReadTokens
{
    public static void main(String[] args) throws Exception
    {
            Scanner sc = new Scanner(new File("text.txt"));
            while (sc.hasNext())
                    System.out.println(sc.next());
            sc.close();
    }
}
```

Output:

Hello
World

---

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

Program

```
import java.io.*;
import java.util.*;
public class WordCount
{
    public static void main(String[] args) throws Exception
    {
            Scanner sc = new Scanner(new File("text.txt"));
            String word = "Hello";
            int count = 0;
            while (sc.hasNext())
            {
                    if (sc.next().equalsIgnoreCase(word))
                    count++;
            }
            sc.close();
            System.out.println("Occurrences of '" + word + "': " + count);
    }
}
```

OutPut:

Occurrences of 'Hello': 1

- **13. Write a program to create, move, and delete a file using Files and Paths.**

Program:

```
import java.nio.file.*;
public class FileOperations
{
    public static void main(String[] args) throws Exception
    {
            Path filePath = Paths.get("test.txt");
            Files.createFile(filePath);
            System.out.println("File created: " + filePath);
            // Move file
            Path newPath = Paths.get("moved_test.txt");
            Files.move(filePath, newPath,
StandardCopyOption.REPLACE_EXISTING);
            System.out.println("File moved to: " + newPath);
            // Delete file
            Files.delete(newPath);
            System.out.println("File deleted.");
    }
}
```

OutPut:

```
File created: test.txt
File moved to: moved_test.txt
File deleted.
```

- **14. Write a program to read all lines of a file using Files.readAllLines() and print them.**

Program:

```
import java.nio.file.*;
import java.util.List;
public class ReadFile
{
    public static void main(String[] args) throws Exception
    {
            Path path = Paths.get("test.txt");
            List<String> lines = Files.readAllLines(path);
            for (String line : lines)
```

```
            {
                System.out.println(line);
            }
    }
}
```

OutPut:

Hello
World

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

Program:

```
import java.io.file.*;
import java.nio.file.StandardOpenOption;
public class SimpleWriteAppend
{
    public static void main(String[] args) throws Exception
    {
            Path file = Paths.get("test.txt");
            // Write
            Files.write(file, "Hello, World!\n".getBytes());
            // Append
                Files.write(file, "This is appended text.\n".getBytes(),
            StandardOpenOption.APPEND);
            System.out.println("Done!");
    }
}
```

OutPut:

Hello, World!
This is appended text.

- **16. Write a program to walk through a directory tree and display file names using Files.walk().**

Program:

```
import java.nio.file.*;
public class WalkDir
{
    public static void main(String[] args) throws Exception
```

```
    {
        Files.walk(Paths.get(".")) // current directory
      .  forEach(System.out::println);
    }
}
```

- **17. Write a program to copy a file using Files.copy() with REPLACE_EXISTING option.**

  Program:

```
import java.nio.file.*;
public class CopyFile
{
    public static void main(String[] args) throws Exception
    {
        Path source = Paths.get("source.txt");
        Path dest = Paths.get("copy.txt");
        Files.copy(source, dest, StandardCopyOption.REPLACE_EXISTING);
        System.out.println("File copied successfully.");
    }
}
```

  OutPut:

```
File copied successfully.
```

- **18. Write a program to check and print the size of a file in bytes using Files.size().**

  Program:

```
import java.io.file.*;
public class FileSize
{
    public static void main(String[] args) throws Exception
    {
        Path path = Paths.get("text.txt");
        long size = Files.size(path);
        System.out.println("File size: " + size + " bytes");
    }
}
```

  Output:

File size: 42 bytes

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

Program:

```
import java.io.*;
class Employee implements Serializable
{
    String name;
    int id;
    Employee(String name, int id)
    {
            this.name = name;
            this.id = id;
    }
}

public class SerializeEmployee
{
    public static void main(String[] args) throws Exception
    {
            Employee emp = new Employee("Jayanth", 1);
            FileOutputStream fos = new FileOutputStream("employee.ser");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(emp);
            oos.close();
            System.out.println("Employee object serialized to employee.ser");
    }
}
```

Output:

Employee object serialized to employee.ser

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

Program:

```
import java.io.*;
class Employee implements Serializable
{
    String name;
```

```
    int id;
}
public class DeserializeEmployee
{
    public static void main(String[] args) throws Exception
    {
            FileInputStream fis = new FileInputStream("employee.ser");
            ObjectInputStream ois = new ObjectInputStream(fis);
            Employee emp = (Employee) ois.readObject();
            ois.close();
            System.out.println("Name: " + emp.name);
            System.out.println("ID: " + emp.id);
    }
}
```

Output:

```
Name: Jayanth
ID: 1
```