Java I/O File Handling -

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

Code; **package** File\_Handling\_practice;

**import** java.io.File;

**import** java.io.IOException;

**public** **class** Creatingatextfile {

**public** **static** **void** main(String[] args) {

File f=**new** File("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\test.txt");

**try** {

**if**(f.createNewFile()){

System.***out***.println("TEST text file is created successfully.");

}**else** {

System.***out***.println("test.txt file is already existed.");

}

}**catch**(IOException e) {

System.***out***.println("An error occured while creating the file:"+e.getMessage());

e.printStackTrace();

}

}

}

Output; test.txt file is already existed.

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

Code; **package** File\_Handling\_practice;

**import** java.io.File;

**import** java.io.IOException;

**public** **class** Tocheckfileexists {

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

{

File f= **new** File("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample.txt");

**if**(f.exists())

{

System.***out***.println("File is exists in the path given "+f.toPath());

}**else** {

System.***out***.println("File is not exists in the given path.");

}

}

}

Output; File is exists in the path given C:\Users\tatip\OneDrive\Desktop\New folder\sample.txt

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

Code: **package** File\_Handling\_practice;

**import** java.io.File;

**import** java.io.IOException;

**import** java.io.FileWriter;

**import** java.io.BufferedWriter;

**public** **class** Towriteintofile {

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

**try**(BufferedWriter f = **new** BufferedWriter(**new** FileWriter("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample.txt"))){

f.write("Hello, World!");

System.***out***.println("Hello world is written to the file.");

}**catch**(IOException e) {

e.printStackTrace();

}

}

}

Output; Hello world is written to the file.

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

Code: **package** File\_Handling\_practice;

**import** java.io.File;

**import** java.io.IOException;

**import** java.io.FileReader;

**import** java.io.BufferedReader;

**public** **class** Toreadafile {

**public** **static** **void** main(String[] args) {

**try** {

BufferedReader bf=**new** BufferedReader(**new** FileReader("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample13.txt"));

String line;

**while**((line=bf.readLine())!=**null**)

{

System.***out***.println(line);

}

}**catch**(IOException e) {

e.printStackTrace();

}

}

}

Output; Hello Welcome

Java sessions are begins.

5. Write a program to append a line of text to an existing file.

Code; package File\_Handling\_practice;

import java.io.File;

import java.io.IOException;

import java.io.FileWriter;

import java.io.BufferedWriter;

public class Towriteintofile {

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

String filePath ="C:\\\\Users\\\\tatip\\\\OneDrive\\\\Desktop\\\\New folder\\\\sample.txt";

String lineToAppend="Hello, World!";

* try(BufferedWriter f = new BufferedWriter(new FileWriter(filePath,true))){
* f.write(lineToAppend);
* System.out.println("Text appended to the file successfully.");
* }catch(IOException e) {
* System.err.println("Error while appending to file:");
* e.printStackTrace();
* }
* }
* }

Output; Text appended to the file successfully.

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**public** **class** Count\_lines\_words\_characters {

**public** **static** **void** main(String[] args) {

String filePath = "C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample.txt";

**try** (BufferedReader reader = **new** BufferedReader(**new** FileReader(filePath))) {

String line;

**int** lineCount=0;

**int** wordCount=0;

**int** charCount=0;

**while** ((line = reader.readLine()) != **null**) {

lineCount++;

// Count characters (excluding newline)

charCount += line.length();

// Split line into words using whitespace

String[] words = line.trim().split("\\s+");

**if** (!line.trim().isEmpty()) {

wordCount += words.length;

}

}

System.***out***.println("Total Lines: " + lineCount);

System.***out***.println("Total Words: " + wordCount);

System.***out***.println("Total Characters: " + charCount);

}**catch**(IOException e ) {

System.***out***.println("Error reading the file:");

e.printStackTrace();

}

}

}

Output; Total Lines: 1

Total Words: 4

Total Characters: 39

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**public** **class** Copy\_onefile\_another {

**public** **static** **void** main(String[] args) {

**try** {

BufferedReader reader=**new** BufferedReader(**new** FileReader("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\source.txt"));

BufferedWriter writer=**new** BufferedWriter(**new** FileWriter("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\destination.txt"));

String line;

**while** ((line = reader.readLine()) != **null**) {

writer.write(line);

writer.newLine();

}

System.***out***.println("File copied successfully.");

reader.close();

writer.close();

}**catch**(IOException e) {

System.***out***.println("Error occured while copying file");

e.printStackTrace();

}

}

}

Output; File copied successfully.

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

Code; package File\_Handling\_practice;

import java.io.\*;

* import java.nio.file.Files;
* import java.util.Scanner;
* public class Print\_all\_files {
* public static void main(String[] args) {
* Scanner scan= new Scanner(System.in);
* System.out.println("Enter folder path:");
* String path=scan.nextLine();
* File folder = new File(path);
* if (folder.isDirectory()) {
* File[] files = folder.listFiles();
* System.out.println("Files inside the folder:");
* for (File file : files) {
* if (file.isFile()) {
* System.out.println(file.getName());
* }
* }
* } else {
* System.out.println("The path entered is not a valid directory.");
* }
* scan.close();
* }

}

Output; Enter folder path:

C:\Users\tatip\OneDrive\Desktop\New folder

Files inside the folder:

data.txt

destination.txt

notes of c.doc

notes of c.docx

notes.txt

report.txt

reserved.txt

reversed.txt

sample.txt

sample1.text

sample1.texts

sample1.txt

sample13.txt

source.txt

story.txt

student.txt

test.txt

updated\_story.txt

userinput.txt

win.docx

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**import** java.util.Scanner;

**public** **class** To\_display\_txt\_files {

**public** **static** **void** main(String[] args) {

Scanner scan= **new** Scanner(System.***in***);

System.***out***.println("Enter folder path:");

String path=scan.nextLine();

File folder = **new** File(path);

**if** (folder.isDirectory()) {

File[] txtFiles = folder.listFiles(*getTxtFileFilter*());

**if** (txtFiles == **null** || txtFiles.length == 0) {

System.***out***.println("No .txt files found in the folder.");

} **else** {

System.***out***.println(".txt files in the folder:");

**for** (File file : txtFiles) {

System.***out***.println("• " + file.getName());

}

}

} **else** {

System.***out***.println("The path entered is not a valid directory.");

}

scan.close();

}

**private** **static** FilenameFilter getTxtFileFilter() {

**return** **new** FilenameFilter() {

@Override

**public** **boolean** accept(File dir, String name) {

**return** name.toLowerCase().endsWith(".txt");

}

};

}

}

Output; Enter folder path:

C:\Users\tatip\OneDrive\Desktop\New folder

.txt files in the folder:

• data.txt

• destination.txt

• notes.txt

• report.txt

• reserved.txt

• reversed.txt

• sample.txt

• sample1.txt

• sample13.txt

• source.txt

• story.txt

• student.txt

• test.txt

• updated\_story.txt

• userinput.txt

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**class** Student **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**int** id;

String name;

**float** marks;

Student(**int** id,String name,**float** marks){

**this**.id=id;

**this**.name=name;

**this**.marks=marks;

}

**public** **int** getid() {

**return** id;

}

**public** String getname() {

**return** name;

}

**public** **float** getmarks() {

**return** marks;

}

**public** String toString() {

**return** "Student{id=" + id + ", name='" + name + "', marks=" + marks + "}";

}

}

**public** **class** Store\_objects\_serialization {

**public** **static** **void** main(String[] args) {

String filePath = "student.ser";

Student s1 = **new** Student(101, "Jagadheeswar", 92.5f);

**try** {

FileOutputStream fos = **new** FileOutputStream("student.ser");

ObjectOutputStream oos = **new** ObjectOutputStream(fos);

oos.writeObject(s1);

oos.close();

fos.close();

System.***out***.println("Student object serialized successfully!");

} **catch** (IOException e) {

System.***out***.println("Serialization failed.");

e.printStackTrace();

}

//Deserialization

**try** (ObjectInputStream ois = **new** ObjectInputStream(**new** FileInputStream(filePath))) {

Student deserializedStudent = (Student) ois.readObject();

System.***out***.println("Student object deserialized successfully!");

System.***out***.println("Deserialized Data: " + deserializedStudent);

} **catch** (IOException | ClassNotFoundException e) {

System.***err***.println("Deserialization failed:");

e.printStackTrace();

}

}

}

Output; Student object serialized successfully!

Student object deserialized successfully!

Deserialized Data: Student{id=101, name='Jagadheeswar', marks=92.5}

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

Code; **package** File\_Handling\_practice;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.util.Scanner;

**public** **class** Toreadfile\_Scanner {

**public** **static** **void** main(String[] args) {

String filePath = "C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample13.txt";

**try** (Scanner scanner = **new** Scanner(**new** File(filePath))) {

System.***out***.println("Tokens found in the file:");

**while** (scanner.hasNext()) {

String token = scanner.next();

System.***out***.println("• " + token);

}

} **catch** (FileNotFoundException e) {

System.***err***.println("File not found: " + filePath);

e.printStackTrace();

}

}

}

Output; Tokens found in the file:

• Hii

• Hello

• Welcome

• Java

• sessions

• are

• begins.

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

Code; package File\_Handling\_practice;

* import java.io.\*;
* import java.util.Scanner;
* public class Word\_search\_file {
* public static void main(String[] args) {
* Scanner scan=new Scanner(System.in);
* System.out.println("Enter a word:");
* String s=scan.nextLine();
* int wordcount=0;
* File file=new File("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\notes.txt");
* boolean found = false;
* if (file.exists()) {
* try (Scanner fileScanner = new Scanner(file)) {
* while (fileScanner.hasNextLine()) {
* String line = fileScanner.nextLine();
* if (line.contains(s)) {
* found = true;
* wordcount++;
* break;
* }
* }
* } catch (FileNotFoundException e) {
* System.out.println("Error reading the file.");
* }
* if (found) {
* System.out.println("The word \"" + s + "\" exists for "+wordcount+" times in the file");
* } else {
* System.out.println("The word \"" + s + "\" was not found in the file.");
* }
* }
* else {
* System.out.println("File 'notes.txt' does not exist.");
* }
* scan.close();
* }
* }

Output; Enter a word:

java

The word "java" exists for 1 times in the file

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**import** java.nio.file.\*;

**public** **class** Create\_move\_delete\_file {

**public** **static** **void** main(String[] args) {

**try**{

Path sourcePath=Paths.*get*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\demo1.txt");

Files.*createFile*(sourcePath);

System.***out***.println("File created successfully at: " + sourcePath);

Path targetPath = Paths.*get*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\Moved\\demo1.txt");

Files.*createDirectories*(targetPath.getParent());

Files.*move*(sourcePath, targetPath, StandardCopyOption.***REPLACE\_EXISTING***);

System.***out***.println("File moved successfully to: " + targetPath);

Files.*delete*(targetPath);

System.***out***.println("File deleted successfully from: " + targetPath);

} **catch** (FileAlreadyExistsException e) {

System.***err***.println("File already exists.");

} **catch** (IOException e) {

System.***err***.println("An error occurred:");

e.printStackTrace();

}

}

}

Output; File created successfully at: C:\Users\tatip\OneDrive\Desktop\New folder\demo1.txt

File moved successfully to: C:\Users\tatip\OneDrive\Desktop\New folder\Moved\demo1.txt

File deleted successfully from: C:\Users\tatip\OneDrive\Desktop\New folder\Moved\demo1.txt

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**import** java.nio.file.Files;

**import** java.nio.file.Path;

**import** java.util.List;

**public** **class** To\_read\_all\_lines {

**public** **static** **void** main(String[] args) {

Path filePath = Path.*of*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample.txt");

**try** {

List<String> lines = Files.*readAllLines*(filePath);

System.***out***.println("Contents of the file:");

**for** (String line : lines) {

System.***out***.println(line);

}

} **catch** (IOException e) {

System.***err***.println("Error reading the file:");

e.printStackTrace();

}

}

}

Output; Contents of the file:

Hello, World!Hello, World!Hello, World!

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**import** java.nio.file.Files;

**import** java.nio.file.Path;

**import** java.nio.file.Paths;

**import** java.nio.file.StandardOpenOption;

**import** java.util.List;

**public** **class** File\_write\_append {

**public** **static** **void** main(String[] args) {

Path filePath = Paths.*get*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample.txt");

List<String> linesToWrite = List.*of*("This is a new line of text.", "Appended using Files.write");

**try** {

Files.*createDirectories*(filePath.getParent());

Files.*write*(filePath, linesToWrite, StandardOpenOption.***CREATE***, StandardOpenOption.***APPEND***);

System.***out***.println("Data written and appended successfully.");

}**catch**(IOException e) {

System.***err***.println("Error writing to file:");

e.printStackTrace();

}

}

}

Output; Data written and appended successfully.

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**import** java.nio.file.\*;

**import** java.util.stream.Stream;

**public** **class** Files\_walk\_tree {

**public** **static** **void** main(String[] args) {

Path rootPath = Paths.*get*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder");

**try** (Stream<Path> walkStream = Files.*walk*(rootPath)) {

System.***out***.println("Files found in the directory tree:");

walkStream

.filter(Files::*isRegularFile*) // Only include files, not directories

.forEach(path -> System.***out***.println("• " + path.toString()));

} **catch** (IOException e) {

System.***err***.println("Error walking through the directory tree:");

e.printStackTrace();

}

}

}

Output; Files found in the directory tree:

• C:\Users\tatip\OneDrive\Desktop\New folder\data.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\demo.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\destination.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\notes of c.doc

• C:\Users\tatip\OneDrive\Desktop\New folder\notes of c.docx

• C:\Users\tatip\OneDrive\Desktop\New folder\notes.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\report.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\reserved.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\reversed.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\sample.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\sample1.text

• C:\Users\tatip\OneDrive\Desktop\New folder\sample1.texts

• C:\Users\tatip\OneDrive\Desktop\New folder\sample1.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\sample13.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\source.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\story.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\student.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\test.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\updated\_story.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\userinput.txt

• C:\Users\tatip\OneDrive\Desktop\New folder\win.docx

17. Write a program to copy a file using Files.copy() with REPLACE\_EXISTING option.

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**import** java.nio.file.\*;

**public** **class** Files\_copy {

**public** **static** **void** main(String[] args) {

Path sourcePath = Paths.*get*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\source.txt");

Path destinationPath = Paths.*get*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\destination.txt");

**try** {

Files.*createDirectories*(destinationPath.getParent());

Files.*copy*(sourcePath, destinationPath, StandardCopyOption.***REPLACE\_EXISTING***);

System.***out***.println("File copied successfully to: " + destinationPath);

}**catch**(IOException e) {

System.***err***.println("Error occurred while copying the file:");

e.printStackTrace();

}

}

}

Output; File copied successfully to: C:\Users\tatip\OneDrive\Desktop\New folder\destination.txt

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**import** java.nio.file.\*;

**public** **class** File\_size {

**public** **static** **void** main(String[] args) {

Path filePath = Paths.*get*("C:\\Users\\tatip\\OneDrive\\Desktop\\New folder\\sample.txt");

**try** {

**long** sizeInBytes = Files.*size*(filePath);

System.***out***.println("File size: " + sizeInBytes + " bytes");

}**catch**(IOException e) {

System.***err***.println("Error retrieving file size:");

e.printStackTrace();

}

}

}

Output; File size: 96 bytes

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**class** Employee1 **implements** Serializable {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** **int** id;

**private** String name;

**private** **double** salary;

**public** Employee1(**int** id, String name, **double** salary) {

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

}

@Override

**public** String toString() {

**return** "Employee{id=" + id + ", name='" + name + "', salary=" + salary + "}";

}

}

**public** **class** Employee\_serialize {

**public** **static** **void** main(String[] args) {

Employee1 emp = **new** Employee1(1001, "Jagadheeswar", 85000.0);

**try** (ObjectOutputStream oos = **new** ObjectOutputStream(**new** FileOutputStream("employee.ser"))){

oos.writeObject(emp);

System.***out***.println("Employee object serialized and stored in employee.ser");

}**catch**(IOException e) {

System.***err***.println("Serialization failed:");

e.printStackTrace();

}

}

}

Output; Employee object serialized and stored in employee.ser

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

Code; **package** File\_Handling\_practice;

**import** java.io.\*;

**class** Employee1 **implements** Serializable {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** **int** id;

**private** String name;

**private** **double** salary;

**public** Employee1(**int** id, String name, **double** salary) {

**this**.id = id;

**this**.name = name;

**this**.salary = salary;

}

@Override

**public** String toString() {

**return** "Employee{id=" + id + ", name='" + name + "', salary=" + salary + "}";

}

}

**public** **class** Employee\_deserialize {

**public** **static** **void** main(String[] args) {

String filePath = "employee.ser";

**try** (ObjectInputStream ois = **new** ObjectInputStream(**new** FileInputStream(filePath))) {

Employee1 emp = (Employee1) ois.readObject();

System.***out***.println("Employee object deserialized successfully!");

System.***out***.println("Deserialized Data: " + emp);

} **catch** (IOException | ClassNotFoundException e) {

System.***err***.println("Deserialization failed:");

e.printStackTrace();

}

}

}

Output; Employee object deserialized successfully!

Deserialized Data: Employee{id=1001, name='Jagadheeswar', salary=85000.0}