**Java I/O File Handling**

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

ANS:  
import java.io.File;

import java.io.IOException;

public class CreateNewFile {

public static void main(String[] args) {

File file = new File("test.txt");

try {

if (file.createNewFile()) {

System.out.println("File created: " + file.getName());

} else {

System.out.println("File already exists.");

}

} catch (IOException e) {

System.out.println("Error creating file: " + e.getMessage());

}

}

}

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

ANS:

import java.io.File;

import java.util.Scanner;

public class CheckFileExists {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter file path: ");

String path = sc.nextLine();

File file = new File(path);

if (file.exists()) {

System.out.println("File exists.");

} else {

System.out.println("File does not exist.");

}

sc.close();

}

}

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

ANS:

import java.io.FileWriter;

import java.io.IOException;

public class WriteHelloWorld {

public static void main(String[] args) {

try (FileWriter writer = new FileWriter("hello.txt")) {

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

} catch (IOException e) {

System.out.println("Error writing to file: " + e.getMessage());

}

}

}

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

ANS:

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class ReadFileLineByLine {

public static void main(String[] args) {

try (BufferedReader br = new BufferedReader(new FileReader("input.txt"))) {

String line;

while ((line = br.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

System.out.println("Error reading file: " + e.getMessage());

}

}

}

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

ANS:

import java.io.FileWriter;

import java.io.IOException;

public class AppendToFile {

public static void main(String[] args) {

try (FileWriter fw = new FileWriter("existingfile.txt", true)) {

fw.write("This is an appended line.\n");

} catch (IOException e) {

System.out.println("Error appending to file: " + e.getMessage());

}

}

}

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

ANS:

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class FileStatistics {

public static void main(String[] args) {

int lines = 0, words = 0, characters = 0;

try (BufferedReader br = new BufferedReader(new FileReader("input.txt"))) {

String line;

while ((line = br.readLine()) != null) {

lines++;

characters += line.length();

words += line.trim().isEmpty() ? 0 : line.trim().split("\\s+").length;

}

} catch (IOException e) {

System.out.println("Error reading file: " + e.getMessage());

}

System.out.println("Lines: " + lines);

System.out.println("Words: " + words);

System.out.println("Characters: " + characters);

}

}

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

ANS:

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class CopyFileContent {

public static void main(String[] args) {

try (FileReader fr = new FileReader("source.txt");

FileWriter fw = new FileWriter("destination.txt")) {

int c;

while ((c = fr.read()) != -1) {

fw.write(c);

}

} catch (IOException e) {

System.out.println("Error: " + e.getMessage());

}

}

}

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

ANS:  
import java.io.File;

public class ListFilesInDirectory {

public static void main(String[] args) {

File dir = new File(".");

File[] files = dir.listFiles();

if (files != null) {

for (File f : files) {

System.out.println(f.getName());

}

}

}

}

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

ANS:  
import java.io.File;

import java.io.FilenameFilter;

public class FilterTxtFiles {

public static void main(String[] args) {

File dir = new File(".");

FilenameFilter txtFilter = (dir1, name) -> name.endsWith(".txt");

String[] txtFiles = dir.list(txtFilter);

if (txtFiles != null) {

for (String file : txtFiles) {

System.out.println(file);

}

}

}

}

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

ANS:

import java.io.\*;

class Student implements Serializable {

private static final long serialVersionUID = 1L;

int id;

String name;

double marks;

public Student(int id, String name, double marks) {

this.id = id;

this.name = name;

this.marks = marks;

}

public String toString() {

return "ID: " + id + ", Name: " + name + ", Marks: " + marks;

}

}

public class SerializeDeserializeStudent {

public static void main(String[] args) {

Student s1 = new Student(101, "Alice", 89.5);

// Serialize

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

oos.writeObject(s1);

} catch (IOException e) {

System.out.println("Serialization error: " + e.getMessage());

}

// Deserialize

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("student.ser"))) {

Student s2 = (Student) ois.readObject();

System.out.println("Deserialized Student: " + s2);

} catch (IOException | ClassNotFoundException e) {

System.out.println("Deserialization error: " + e.getMessage());

}

}

}

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

ANS:  
import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class ReadFileTokens {

public static void main(String[] args) {

try (Scanner sc = new Scanner(new File("input.txt"))) {

while (sc.hasNext()) {

System.out.println(sc.next());

}

} catch (FileNotFoundException e) {

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

}

}

}

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

ANS:  
import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class WordCountInFile {

public static void main(String[] args) {

String searchWord = "Java";

int count = 0;

try (BufferedReader br = new BufferedReader(new FileReader("input.txt"))) {

String line;

while ((line = br.readLine()) != null) {

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

for (String w : words) {

if (w.equals(searchWord)) {

count++;

}

}

}

} catch (IOException e) {

System.out.println("Error: " + e.getMessage());

}

System.out.println("The word '" + searchWord + "' occurs " + count + " times.");

}

}

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

ANS:  
import java.io.IOException;

import java.nio.file.\*;

public class FileOperations {

public static void main(String[] args) {

Path source = Paths.get("testfile.txt");

Path target = Paths.get("movedfile.txt");

try {

Files.createFile(source);

Files.move(source, target, StandardCopyOption.REPLACE\_EXISTING);

Files.delete(target);

System.out.println("File created, moved, and deleted successfully.");

} catch (IOException e) {

System.out.println("Error: " + e.getMessage());

}

}

}

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

ANS:

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.util.List;

public class ReadAllLinesExample {

public static void main(String[] args) {

Path file = Path.of("input.txt");

try {

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

for (String line : lines) {

System.out.println(line);

}

} catch (IOException e) {

System.out.println("Error reading file: " + e.getMessage());

}

}

}

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

ANS:

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.StandardOpenOption;

import java.util.List;

public class WriteAndAppendFile {

public static void main(String[] args) {

Path file = Path.of("output.txt");

List<String> linesToWrite = List.of("Line 1", "Line 2");

List<String> linesToAppend = List.of("Appended Line 1", "Appended Line 2");

try {

Files.write(file, linesToWrite);

Files.write(file, linesToAppend, StandardOpenOption.APPEND);

System.out.println("Write and append completed.");

} catch (IOException e) {

System.out.println("Error: " + e.getMessage());

}

}

}

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

ANS:

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.util.stream.Stream;

public class DirectoryWalk {

public static void main(String[] args) {

Path start = Path.of(".");

try (Stream<Path> stream = Files.walk(start)) {

stream.filter(Files::isRegularFile)

.forEach(path -> System.out.println(path));

} catch (IOException e) {

System.out.println("Error walking directory: " + e.getMessage());

}

}

}

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

ANS:

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.StandardCopyOption;

public class CopyFileExample {

public static void main(String[] args) {

Path source = Path.of("source.txt");

Path destination = Path.of("destination.txt");

try {

Files.copy(source, destination, StandardCopyOption.REPLACE\_EXISTING);

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

} catch (IOException e) {

System.out.println("Copy failed: " + e.getMessage());

}

}

}

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

ANS:

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

public class FileSizeCheck {

public static void main(String[] args) {

Path file = Path.of("example.txt");

try {

long size = Files.size(file);

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

} catch (IOException e) {

System.out.println("Error getting file size: " + e.getMessage());

}

}

}

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

ANS:

import java.io.\*;

class Employee implements Serializable {

private static final long serialVersionUID = 1L;

int id;

String name;

double salary;

Employee(int id, String name, double salary) {

this.id = id;

this.name = name;

this.salary = salary;

}

}

public class SerializeEmployee {

public static void main(String[] args) {

Employee emp = new Employee(1001, "John Doe", 75000.0);

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

oos.writeObject(emp);

System.out.println("Employee serialized successfully.");

} catch (IOException e) {

System.out.println("Serialization failed: " + e.getMessage());

}

}

}

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

ANS:

import java.io.\*;

class Employee implements Serializable {

private static final long serialVersionUID = 1L;

int id;

String name;

double salary;

}

public class DeserializeEmployee {

public static void main(String[] args) {

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("employee.ser"))) {

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

System.out.println("ID: " + emp.id);

System.out.println("Name: " + emp.name);

System.out.println("Salary: " + emp.salary);

} catch (IOException | ClassNotFoundException e) {

System.out.println("Deserialization failed: " + e.getMessage());

}

}

}