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

**Code:**

import java.io.FileWriter;

import java.io.IOException;

public class CreateTextFile {

public static void main(String[] args) {

try {

FileWriter writer = new FileWriter("test.txt");

writer.write("Hello! This is a new text file.\n");

writer.write("Java file writing example.\n");

writer.close();

System.out.println("test.txt has been created successfully!");

}

catch (IOException e) {

System.out.println("An error occurred while creating the file.");

e.printStackTrace();

}

}

}

**Output:**

test.txt has been created successfully!

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

**Code:**

import java.io.File;

import java.util.Scanner;

public class CheckFileExists {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String filePath = scanner.nextLine();

File file = new File(filePath);

if (file.exists()) {

System.out.println(" The file exists at: " + filePath);

} else {

System.out.println("The file does NOT exist at: " + filePath);

}

scanner.close();

}

}

**Output:**

Enter the file path: C:\Users\Yasawini\Documents\test.txt The file exists at: C:\Users\Yasawini\Documents\test.txt

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

**Code:**

import java.io.FileWriter; // To write into a file

import java.io.IOException; // To handle possible errors

public class WriteHelloWorld {

public static void main(String[] args) {

try {

FileWriter writer = new FileWriter("hello.txt");

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

writer.close();

System.out.println("hello.txt has been created and text written successfully!");

}

catch (IOException e) {

System.out.println("An error occurred while writing to the file.");

e.printStackTrace();

}

}

}

**Output:**

hello.txt has been created and text written successfully!

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

**Code:**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class ReadFileExample {

public static void main(String[] args) {

String fileName = "test.txt";

try {

BufferedReader reader = new BufferedReader(new FileReader(fileName));

String line;

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

System.out.println(line);

}

reader.close();

}

catch (IOException e) {

System.out.println("An error occurred while reading the file.");

e.printStackTrace();

}

}

}

**Output:**

Hello! This is the first line.

This is the second line.

And here is the third line.

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

**Code:**

import java.io.FileWriter;

import java.io.IOException;

public class AppendToFile {

public static void main(String[] args) {

String fileName = "test.txt";

try {

FileWriter writer = new FileWriter(fileName, true);

writer.write("\nThis is an appended line.");

writer.close();

System.out.println("Text has been appended to " + fileName);

}

catch (IOException e) {

System.out.println("An error occurred while appending to the file.");

e.printStackTrace();

}

}

}

**Output:**

Hello! This is a new text file.

Java file writing example.

This is an appended line.

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

**Code:**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class FileCountExample {

public static void main(String[] args) {

String fileName = "test.txt";

int lineCount = 0;

int wordCount = 0;

int charCount = 0;

try {

BufferedReader reader = new BufferedReader(new FileReader(fileName));

String line;

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

lineCount++;

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

wordCount += words.length;

charCount += line.length();

}

reader.close();

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

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

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

} catch (IOException e) {

System.out.println("An error occurred.");

e.printStackTrace();

}

}

}

**test.txt**

Hello World

Java is fun

I love programming

**Output:**

Lines: 3

Words: 7

Characters: 39

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

**Code:**

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class CopyFileExample {

public static void main(String[] args) {

String sourceFile = "source.txt";

String destinationFile = "destination.txt";

try {

FileReader reader = new FileReader(sourceFile);

FileWriter writer = new FileWriter(destinationFile);

int ch;

while ((ch = reader.read()) != -1) {

writer.write(ch);

}

reader.close();

writer.close();

System.out.println("File copied successfully from " + sourceFile + " to " + destinationFile);

} catch (IOException e) {

System.out.println("An error occurred while copying the file.");

e.printStackTrace();

}

}

}

**Output:**

File copied successfully from source.txt to destination.txt

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

**Code:**

import java.io.File;

import java.util.Scanner;

public class ListFilesExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the directory path: ");

String directoryPath = scanner.nextLine();

File directory = new File(directoryPath);

if (directory.isDirectory()) {

String[] files = directory.list();

if (files != null && files.length > 0) {

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

for (String fileName : files) {

System.out.println(fileName);

}

} else {

System.out.println("The directory is empty.");

}

} else {

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

}

scanner.close();

}

}

**Input:**

Enter the directory path: C:\Users\yasaswini\Documents

**Output:**

Files in the directory:

notes.txt

project.docx

image.png

java\_programs

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

**Code:**

import java.io.File;

import java.io.FilenameFilter;

import java.util.Scanner;

public class TxtFileFilterExample {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the folder path: ");

String folderPath = scanner.nextLine();

File folder = new File(folderPath);

if (folder.isDirectory()) {

FilenameFilter txtFilter = new FilenameFilter() {

public boolean accept(File dir, String name) {

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

}

};

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

if (txtFiles != null && txtFiles.length > 0) {

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

for (String fileName : txtFiles) {

System.out.println(fileName);

}

} else {

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

}

} else {

System.out.println("The path you entered is not a valid folder.");

}

scanner.close();

}

}

**Input:**

Enter the folder path: C:\Users\yasawini\Documents

**Output:**

Text files in the folder:

notes.txt

tasks.txt

test.txt

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

**code:**

import java.io.\*;

class Student implements Serializable {

String name;

int age;

String grade;

Student(String name, int age, String grade) {

this.name = name;

this.age = age;

this.grade = grade;

}

}

public class StudentSerializationExample {

public static void main(String[] args) {

String fileName = "student.ser";

Student student = new Student("John Doe", 20, "A");

try {

FileOutputStream fileOut = new FileOutputStream(fileName);

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(student);

out.close();

fileOut.close();

System.out.println("Student object has been serialized and saved to " + fileName);

} catch (IOException e) {

System.out.println("Error during serialization.");

e.printStackTrace();

}

try {

FileInputStream fileIn = new FileInputStream(fileName);

ObjectInputStream in = new ObjectInputStream(fileIn);

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

in.close();

fileIn.close();

System.out.println("Student object has been deserialized:");

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

System.out.println("Age: " + deserializedStudent.age);

System.out.println("Grade: " + deserializedStudent.grade);

} catch (IOException | ClassNotFoundException e) {

System.out.println("Error during deserialization.");

e.printStackTrace();

}

}

}

**Output:**

Student object has been serialized and saved to student.ser

Student object has been deserialized:

Name: John Doe

Age: 20

Grade: A

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

**Code:**

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class ReadFileTokens {

public static void main(String[] args) {

try {

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

Scanner scanner = new Scanner(file);

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

while (scanner.hasNext()) {

String token = scanner.next();

System.out.println(token);

}

scanner.close();

} catch (FileNotFoundException e) {

System.out.println("File not found.");

}

}

}

**Test.txt**

Hello world

This is Java

**Output:**

Tokens in the file:

Hello

world

This

is

Java

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

**Code:**

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class WordSearchInFile {

public static void main(String[] args) {

Scanner inputScanner = new Scanner(System.in);

System.out.print("Enter file name (with path if needed): ");

String fileName = inputScanner.nextLine();

System.out.print("Enter word to search: ");

String wordToFind = inputScanner.next();

int count = 0;

try {

File file = new File(fileName);

Scanner fileScanner = new Scanner(file);

while (fileScanner.hasNext()) {

String token = fileScanner.next();

if (token.equalsIgnoreCase(wordToFind)) {

count++;

}

}

fileScanner.close();

System.out.println("The word \"" + wordToFind + "\" occurred " + count + " times.");

} catch (FileNotFoundException e) {

System.out.println("File not found.");

}

inputScanner.close();

}

}

**Test.txt**

Java is fun. I love learning Java because Java is powerful.

**Output:**

Enter file name (with path if needed): test.txt

Enter word to search: Java

The word "Java" occurred 3 times.

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

**Code:**

import java.io.IOException;

import java.nio.file.\*;

public class FileOperationsExample {

public static void main(String[] args) {

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

Path newPath = Paths.get("moved\_testfile.txt");

try {

Files.createFile(originalPath);

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

Files.move(originalPath, newPath, StandardCopyOption.REPLACE\_EXISTING);

System.out.println("File moved to: " + newPath);

Files.delete(newPath);

System.out.println("File deleted: " + newPath);

} catch (FileAlreadyExistsException e) {

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

} catch (NoSuchFileException e) {

System.out.println("File not found.");

} catch (IOException e) {

System.out.println("An error occurred.");

e.printStackTrace();

}

}

}

**Output:**

File created: testfile.txt

File moved to: moved\_testfile.txt

File deleted: moved\_testfile.txt

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

**Code:**

import java.io.IOException;

import java.nio.file.\*;

import java.util.List;

public class ReadAllLinesExample {

public static void main(String[] args) {

Path filePath = Paths.get("test.txt");

try {

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

System.out.println("File content:");

for (String line : lines) {

System.out.println(line);

}

} catch (IOException e) {

System.out.println("An error occurred while reading the file.");

}

}

}

**Output:**

File content:

Hello World

Java is fun

Learning file handling

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

**Code:**

import java.io.IOException;

import java.nio.file.\*;

import java.util.Arrays;

public class WriteAndAppendExample {

public static void main(String[] args) {

Path filePath = Paths.get("example.txt");

try {

Files.write(filePath, Arrays.asList("Hello, World!", "This is Java NIO."));

System.out.println("Data written to file.");

Files.write(filePath, Arrays.asList("Appending this line."), StandardOpenOption.APPEND);

System.out.println("Data appended to file.");

} catch (IOException e) {

System.out.println("An error occurred while writing to the file.");

}

}

}

**Output:**

Data written to file.

Data appended to file.

**Example.txt**

Hello, World!

This is Java NIO.

Appending this line.

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

**Code:**

import java.io.IOException;

import java.nio.file.\*;

import java.util.stream.Stream;

public class WalkDirectoryExample {

public static void main(String[] args) {

Path startPath = Paths.get("."); // Current directory

try (Stream<Path> paths = Files.walk(startPath)) {

System.out.println("Files and folders:");

paths.forEach(System.out::println);

} catch (IOException e) {

System.out.println("An error occurred while walking through the directory.");

}

}

}

**Output:**

Files and folders:

.

./test.txt

./example.txt

./subfolder

./subfolder/data.txt

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

**Code:**

import java.io.IOException;

import java.nio.file.\*;

public class CopyFileExample {

public static void main(String[] args) {

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

Path destinationPath = Paths.get("destination.txt");

try {

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

System.out.println("File copied successfully from " + sourcePath + " to " + destinationPath);

} catch (IOException e) {

System.out.println("An error occurred while copying the file.");

}

}

}

**Output:**

File copied successfully from source.txt to destination.txt

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

**Code:**

import java.io.IOException;

import java.nio.file.\*;

public class FileSizeExample {

public static void main(String[] args) {

Path filePath = Paths.get("test.txt"); // Change file name if needed

try {

long size = Files.size(filePath);

System.out.println("Size of the file: " + size + " bytes");

} catch (IOException e) {

System.out.println("An error occurred while checking the file size.");

}

}

}

**Output:**

Size of the file: 42 bytes

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

**Code:**

import java.io.\*;

class Employee implements Serializable {

String name;

int age;

double salary;

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

this.name = name;

this.age = age;

this.salary = salary;

}

}

public class SerializeEmployee {

public static void main(String[] args) {

Employee emp = new Employee("John Doe", 30, 50000.0);

try (FileOutputStream fileOut = new FileOutputStream("employee.ser");

ObjectOutputStream out = new ObjectOutputStream(fileOut)) {

out.writeObject(emp);

System.out.println("Employee object has been serialized to employee.ser");

} catch (IOException e) {

System.out.println("An error occurred during serialization.");

}

}

}

**Output:**

Employee object has been serialized to employee.ser

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

**Code:**

import java.io.\*;

class Employee implements Serializable {

String name;

int age;

double salary;

}

public class DeserializeEmployee {

public static void main(String[] args) {

try (FileInputStream fileIn = new FileInputStream("employee.ser");

ObjectInputStream in = new ObjectInputStream(fileIn)) {

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

System.out.println("Employee object has been deserialized:");

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

System.out.println("Age: " + emp.age);

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

} catch (IOException | ClassNotFoundException e) {

System.out.println("An error occurred during deserialization.");

}

}

}

**Output:**

Employee object has been deserialized:

Name: John Doe

Age: 30

Salary: 50000.0