**1.**

package Practiceday10;

import java.io.\*;

import java.nio.file.\*;

import java.util.Scanner;

public class Problem1{

public static void main(String[] args) {

if (args.length != 2) {

System.out.println("Usage: java FileCopy <srcfile> <destifile>");

return;

}

String srcfile = args[0];

String destifile = args[1];

Path sourcePath = Paths.get(srcfile);

Path destinationPath = Paths.get(destifile);

if (!Files.exists(sourcePath)) {

System.out.println("Source file does not exist: " + srcfile);

return;

}

if (Files.exists(destinationPath)) {

System.out.println("Destination file already exists: " + destifile);

Scanner scanner = new Scanner(System.in);

System.out.print("Do you want to overwrite it? (Yes/No): ");

String userResponse = scanner.nextLine().trim().toLowerCase();

if (!userResponse.equals("yes")) {

System.out.println("Operation aborted by the user.");

scanner.close();

return;

}

} else {

try {

Files.createFile(destinationPath);

} catch (IOException e) {

System.out.println("Failed to create destination file: " + e.getMessage());

return;

}

}

try (InputStream inputStream = new FileInputStream(srcfile);

OutputStream outputStream = new FileOutputStream(destifile)) {

byte[] buffer = new byte[1024];

int bytesRead;

while ((bytesRead = inputStream.read(buffer)) != -1) {

outputStream.write(buffer, 0, bytesRead);

}

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

} catch (IOException e) {

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

}

}

}

**2.**

package Practiceday10;

import java.io.\*;

import java.util.\*;

import java.util.stream.Stream;

class Student implements Serializable {

private int rollNo;

private String name;

private int age;

private String address;

public Student(int rollNo, String name, int age, String address) {

this.rollNo = rollNo;

this.name = name;

this.age = age;

this.address = address;

}

@Override

public String toString() {

return "Student{" +

"rollNo=" + rollNo +

", name='" + name + '\'' +

", age=" + age +

", address='" + address + '\'' +

'}';

}

}

class CustomException extends Exception {

public CustomException(String message) {

super(message);

}

}

public class Problem2 {

private static final String FILE\_NAME = "student\_records.ser";

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<Student> students = new ArrayList<>();

while (true) {

try {

System.out.print("Enter Roll Number: ");

String rollNoInput = scanner.nextLine();

if (rollNoInput.isBlank()) throw new CustomException("Roll Number cannot be blank.");

int rollNo = Integer.parseInt(rollNoInput);

System.out.print("Enter Name: ");

String name = scanner.nextLine();

if (name.isBlank()) throw new CustomException("Name cannot be blank.");

System.out.print("Enter Age: ");

String ageInput = scanner.nextLine();

if (ageInput.isBlank()) throw new CustomException("Age cannot be blank.");

int age = Integer.parseInt(ageInput);

System.out.print("Enter Address: ");

String address = scanner.nextLine();

if (address.isBlank()) throw new CustomException("Address cannot be blank.");

students.add(new Student(rollNo, name, age, address));

System.out.print("Do you want to write the data in the file? (Yes/No): ");

String userResponse = scanner.nextLine().trim().toLowerCase();

if (userResponse.equals("yes")) {

writeToFile(students);

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

} else {

System.out.println("Program terminated without saving data.");

break;

}

} catch (NumberFormatException e) {

System.out.println("Roll Number and Age should be numeric.");

} catch (CustomException e) {

System.out.println(e.getMessage());

}

}

scanner.close();

}

private static void writeToFile(List<Student> students) {

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(FILE\_NAME, true))) {

for (Student student : students) {

oos.writeObject(student);

}

} catch (IOException e) {

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

}

}

}

**3.**

**package** Practiceday10;

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

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** Problem3 {

**private** **static** **final** String ***FILE\_NAME*** = "student\_records.ser";

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

List<Student> students = *readFromFile*();

**if** (students.isEmpty()) {

System.***out***.println("No records found.");

} **else** {

students.forEach(System.***out***::println);

}

}

**private** **static** List<Student> readFromFile() {

List<Student> students = **new** ArrayList<>();

**try** (ObjectInputStream ois = **new** ObjectInputStream(**new** FileInputStream(***FILE\_NAME***))) {

**while** (**true**) {

**try** {

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

students.add(student);

} **catch** (EOFException e) {

**break**;

}

}

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

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

}

**return** students;

}

}

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

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

**private** **int** rollNo;

**private** String name;

**private** **int** age;

**private** String address;

**public** Student(**int** rollNo, String name, **int** age, String address) {

**this**.rollNo = rollNo;

**this**.name = name;

**this**.age = age;

**this**.address = address;

}

@Override

**public** String toString() {

**return** "Student{" +

"rollNo=" + rollNo +

", name='" + name + '\'' +

", age=" + age +

", address='" + address + '\'' +

'}';

}

}

**4.**

**package** Practiceday10;

**import** java.io.IOException;

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

**import** java.util.Scanner;

**public** **class** Problem4 {

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

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

System.***out***.println("Enter the source file path:");

String sourcePathString = scanner.nextLine();

System.***out***.println("Enter the destination file path:");

String destinationPathString = scanner.nextLine();

Path sourcePath = Paths.*get*(sourcePathString);

Path destinationPath = Paths.*get*(destinationPathString);

**try** {

**if** (!Files.*exists*(sourcePath)) {

System.***out***.println("Source file does not exist.");

**return**;

}

**if** (Files.*exists*(destinationPath)) {

System.***out***.println("Destination file already exists. Do you want to overwrite it? (Yes/No)");

String response = scanner.nextLine();

**if** (!response.equalsIgnoreCase("Yes")) {

System.***out***.println("Operation cancelled by user.");

**return**;

}

}

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

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

} **catch** (IOException e) {

System.***out***.println("An error occurred during the file copy operation: " + e.getMessage());

}

}

}

5.

**package** Practiceday10;

import java.io.\*;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.util.Scanner;

public class Problem5 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter a string to write to the file:");

String inputString = scanner.nextLine();

String fileName = "io.txt";

try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName))) {

writer.write(inputString);

System.out.println("String written to " + fileName);

} catch (IOException e) {

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

return;

}

try {

Path filePath = Paths.get(fileName);

long fileSize = Files.size(filePath);

System.out.println("Size of " + fileName + ": " + fileSize + " bytes");

} catch (IOException e) {

System.out.println("An error occurred while getting the file size: " + e.getMessage());

return;

}

try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {

String line;

System.out.println("Contents of " + fileName + ":");

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

System.out.println(line);

}

} catch (IOException e) {

System.out.println("An error occurred while reading from the file: " + e.getMessage());

return;

}

File file = new File(fileName);

if (file.delete()) {

System.out.println(fileName + " deleted successfully.");

} else {

System.out.println("Failed to delete " + fileName);

}

}

}

6.

**package** Practiceday10;

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

**public** **class** Problem6 {

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

String[] inputFiles = {"DavidEnglish.txt", "DavidScience.txt", "DavidComputer.txt"};

String outputFile = "DavidNotes.txt";

**try** (BufferedWriter writer = **new** BufferedWriter(**new** FileWriter(outputFile))) {

**for** (String inputFile : inputFiles) {

*writeNotesToOutput*(writer, inputFile);

}

writer.write("\n");

**for** (String inputFile : inputFiles) {

*writeNotesContentToOutput*(writer, inputFile);

}

System.***out***.println("Notes have been compiled into " + outputFile);

} **catch** (IOException e) {

System.***out***.println("An error occurred: " + e.getMessage());

}

}

**private** **static** **void** writeNotesToOutput(BufferedWriter writer, String inputFile) **throws** IOException {

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

writer.write(inputFile);

writer.write("\n");

String line;

**int** lineNumber = 1;

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

writer.write(lineNumber++ + " " + line);

writer.write("\n");

}

writer.write("\n");

}

}

**private** **static** **void** writeNotesContentToOutput(BufferedWriter writer, String inputFile) **throws** IOException {

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

String line;

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

writer.write(line);

writer.write(" ");

}

}

}

}