**Problem Statement 1: Input & Output Streams**

**1. Write a program which take source file and destination file as input as a command line argument. It copies the source file contents to destination file. If source file does not exist, it should give appropriate message to the**

**user. If destination file does not exist, it should be created. If it exists, program should ask that, “whether you want to overwrite? (Yes/No)”.**

**a. Based on user choice, appropriate actions hould be taken.**

**b. Note: Files may be any type of files like bitmap files, exe files, text files etc.**

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.Scanner;

public class Main {

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

Scanner sc=new Scanner(System.in);

String inputfile="C:\\Users\\epavan.reddy\\Desktop\\test\\inputtest1.txt";

String outputfile="C:\\Users\\epavan.reddy\\Desktop\\test\\outputtest2.txt";

File file=new File(outputfile);

if(file.exists()) {

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

System.out.println("Whether you want to overwrite? (true/false)");

boolean agree=true;

boolean result = sc.nextBoolean();

if(agree == result) {

System.out.println("Permission granted");

copyfile(inputfile,outputfile);

}

}else {

System.out.println("Destination file is not exists so we are creating ?(true/false");

boolean check=true;

boolean checks=sc.nextBoolean();

if(checks ==check) {

copyfile(inputfile,outputfile);

}

}

}

public static void copyfile(String filename, String outputfile ) throws IOException

{

try(FileInputStream file1=new FileInputStream(filename);

FileOutputStream output= new FileOutputStream(outputfile)

){

int data;

while((data=file1.read())!=-1) {

output.write(data);

}

System.out.println(" ");

}

catch(FileNotFoundException ex) {

System.out.println("Source file is not in path");

}

}

}

**Output:**

**C:\Users\epavan.reddy>cd desktop**

**C:\Users\epavan.reddy\Desktop>cd test**

**C:\Users\epavan.reddy\Desktop\test>javac Main.java**

**C:\Users\epavan.reddy\Desktop\test>java Main**

**File Exists**

**Whether you want to overwrite? (true/false)**

**true**

**Permission granted**

**C:\Users\epavan.reddy\Desktop\test>**

**2. Write a stream-based program which will accept Roll Number, Name, Age and Address from user. Perform following tasks,**

**a. Age and Roll-no should be numeric.**

**b. Handle built-in exception as None of the field should be blank. Handle with custom exception**

**c. Ask user ,whether to write the data in the file. If answer is yes, then data is saved into a file as an object. (User can write many records in the file), otherwise terminate the current program.**

**d. Hint : Use Serialization.**

import java.io.\*;

import java.util.Scanner;

class BlankFieldException extends Exception {

public BlankFieldException(String message) {

super(message);

}

}

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 "Roll No: " + rollNo + ", Name: " + name + ", Age: " + age + ", Address: " + address;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

File file = new File("students.dat");

while (true) {

try {

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

String rollNoInput = scanner.nextLine();

if (rollNoInput.trim().isEmpty()) {

throw new BlankFieldException("Roll Number cannot be blank.");

}

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

String name = scanner.nextLine();

if (name.trim().isEmpty()) {

throw new BlankFieldException("Name cannot be blank.");

}

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

String ageInput = scanner.nextLine();

if (ageInput.trim().isEmpty()) {

throw new BlankFieldException("Age cannot be blank.");

}

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

String address = scanner.nextLine();

if (address.trim().isEmpty()) {

throw new BlankFieldException("Address cannot be blank.");

}

int rollNo = Integer.parseInt(rollNoInput);

int age = Integer.parseInt(ageInput);

Student student = new Student(rollNo, name, age, address);

System.out.println("Student Record: " + student);

System.out.print("Do you want to save this record to the file? (yes/no): ");

String saveChoice = scanner.nextLine();

if (saveChoice.equalsIgnoreCase("yes")) {

saveToFile(file, student);

}

System.out.print("Do you want to enter another student record? (yes/no): ");

String continueChoice = scanner.nextLine();

if (continueChoice.equalsIgnoreCase("no")) {

break;

}

} catch (BlankFieldException e) {

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

} catch (NumberFormatException e) {

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

} catch (IOException e) {

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

}

}

}

public static void saveToFile(File file, Student student) throws IOException {

boolean append = file.exists();

try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(file, append))) {

if (!append) {

out.writeObject("Student Records:");

}

out.writeObject(student);

System.out.println("Student record saved successfully.");

}

}

}  
**output:**

Microsoft Windows [Version 10.0.19045.5247]

(c) Microsoft Corporation. All rights reserved.

C:\Users\epavan.reddy>**cd desktop**

C:\Users\epavan.reddy\Desktop>**cd test**

C:\Users\epavan.reddy\**Desktop\test>javac Main.java**

C:\Users\epavan.reddy\**Desktop\test>java Main**

**Enter Roll Number: 101**

**Enter Name: sai**

**Enter Age: 23**

**Enter Address: tirupati**

**Student Record: Roll No: 101, Name: sai, Age: 23, Address: tirupati**

**Do you want to save this record to the file? (yes/no): yes**

**Student record saved successfully.**

**Do you want to enter another student record? (yes/no): no**

C:\Users\epavan.reddy\Desktop\test>

**3. Write another classto display all the records savedinto the file in the problem 2**

import java.io.\*;

import java.util.\*;

public class DisplayRecords {

public static void main(String[] args) {

List<Student> students = readDataFromFile();

if (students != null && !students.isEmpty()) {

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

} else {

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

}

}

private static List<Student> readDataFromFile() {

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

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("students.dat"))) {

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;

}

}

**Output:**

C:\Users\epavan.reddy>**cd desktop**

C:\Users\epavan.reddy\**Desktop>cd test**

C:\Users\epavan.reddy\**Desktop\test>javac Main.java**

C:\Users\epavan.reddy\**Desktop\test>java Main**

**Enter Roll Number: 101**

**Enter Name: sai**

**Enter Age: 23**

**Enter Address: Tirupati**

**Student Record: Roll No: 101, Name: sai, Age: 23, Address: Tirupati**

**Do you want to save this record to the file? (yes/no): yes**

**Student record saved successfully.**

**Do you want to enter another student record? (yes/no): no**

C:\Users\epavan.reddy\**Desktop\test>javac DisplayRecords.java**

C:\Users\epavan.reddy\**Desktop\test>java DisplayRecords**

**Roll No: 101, Name: sai, Age: 23, Address: Tirupati**

**4. Write a program using java file system to copy the contents of one file into another. (Refer Java API documentation).**

import java.io.IOException;

import java.nio.file.\*;

public class FileCopy {

public static void main(String[] args) {

String sourceFilePath = "C:\\Users\\epavan.reddy\\Desktop\\test\\inputtest1.txt";

String destinationFilePath ="C:\\Users\\epavan.reddy\\Desktop\\test\\outputtest2.txt";

Path sourcePath = Paths.get(sourceFilePath);

Path destinationPath = Paths.get(destinationFilePath);

try {

if (Files.notExists(sourcePath)) {

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

return;

}

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

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

} catch (IOException e) {

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

}

}

}

**Output:**

C:\Users\epavan.reddy\**Desktop\test>javac FileCopy.java**

C:\Users\epavan.reddy\**Desktop\test>java FileCopy**

**File copied successfully!**

**5. Write a program which will accept an input String from user and do following steps,**

**a. Write the String input in the file io.txt.**

**b. Show size of the file.**

**c. Read contents from the file and display them on console.**

**d. Delete io.txt file using File class.**

import java.io.File;

import java.io.FileWriter;

import java.io.FileReader;

import java.io.BufferedReader;

import java.io.IOException;

import java.util.Scanner;

public class FileOperations {

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 filePath = "io.txt";

try (FileWriter writer = new FileWriter(filePath)) {

writer.write(inputString);

System.out.println("String written to file successfully!");

} catch (IOException e) {

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

return;

}

File file = new File(filePath);

if (file.exists()) {

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

} else {

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

return;

}

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

String line;

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

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

System.out.println(line);

}

} catch (IOException e) {

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

return;

}

if (file.delete()) {

System.out.println("File 'io.txt' deleted successfully.");

} else {

System.out.println("Failed to delete the file.");

}

}

}

**Output:**

C:\Users\epavan.reddy>**cd desktop**

C:\Users\epavan.reddy\**Desktop>cd test**

C:\Users\epavan.reddy\**Desktop\test>javac FileOperations.java**

C:\Users\epavan.reddy\**Desktop\test>java FileOperations**

**Enter a string to write to the file:**

**this is test file**

**String written to file successfully!**

**File size: 17 bytes**

**Contents of the file:**

**this is test file**

**File 'io.txt' deleted successfully.**

**6. For a student David there is a text file named DavidEnglish.txt and DavidScience.txt containing notes for two subjects English, and Science for Semester 1. Similarly, for Semester 2 there is another text file**

**DavidComputer.txt containing notes for Computers. Read these files and write it to a file called DavidNotes.txt having notes for the complete year.**

**Use the following classes for reading and writing to/from the files:**

**• BufferedReader**

**• BufferedWriter**

import java.io.\*;

public class StudentNotes {

public static void main(String[] args) {

String englishFilePath = "DavidEnglish.txt";

String scienceFilePath = "DavidScience.txt";

String computerFilePath = "DavidComputer.txt";

String outputFilePath = "DavidNotes.txt";

try (

BufferedReader englishReader = new BufferedReader(new FileReader(englishFilePath));

BufferedReader scienceReader = new BufferedReader(new FileReader(scienceFilePath));

BufferedReader computerReader = new BufferedReader(new FileReader(computerFilePath));

BufferedWriter writer = new BufferedWriter(new FileWriter(outputFilePath))

) {

writer.write("David's Semester 1 Notes:\n\nEnglish Notes:\n");

String line;

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

writer.write(line);

writer.newLine();

}

writer.newLine();

writer.write("Science Notes:\n");

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

writer.write(line);

writer.newLine();

}

writer.newLine();

writer.write("David's Semester 2 Notes:\n\nComputer Notes:\n");

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

writer.write(line);

writer.newLine();

}

System.out.println("Notes successfully written to DavidNotes.txt");

} catch (IOException e) {

System.out.println("An error occurred while reading or writing files: " + e.getMessage());

}

}

}

**Output:**

C:\Users\epavan.reddy\**Desktop\test>javac StudentNotes.java**

C:\Users\epavan.reddy\**Desktop\test>java StudentNotes**

**Notes successfully written to DavidNotes.txt**

**David's Semester 1 Notes:**

**English Notes:**

**DavidEnglish**

**Science Notes:**

**DavidScience**

**David's Semester 2 Notes:**

**Computer Notes:**

**DavidComputer**