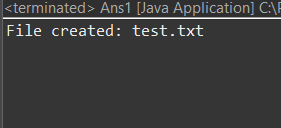
# 1. Write a program to create a new text file named test.txt.

import java.io.File;  
import java.io.IOException;  
  
public class Ans1 {  
 public static void main(String[] args) {  
 try {  
 File file = new File("test.txt");  
 if (file.createNewFile()) {  
 System.out.println("File created: " + file.getName());  
 } else {  
 System.out.println("File already exists.");  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

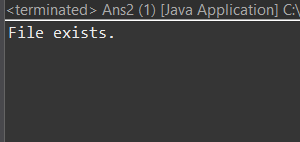
**OUTPUT:**



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

import java.io.File;  
  
public class Ans2 {  
 public static void main(String[] args) {  
 File file = new File("test.txt");  
 if (file.exists()) {  
 System.out.println("File exists.");  
 } else {  
 System.out.println("File does not exist.");  
 }  
 }  
}

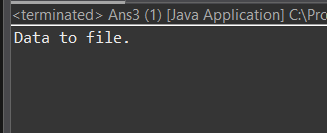
**OUTPUT:**



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

import java.io.FileWriter;  
import java.io.IOException;  
  
public class Ans3 {  
 public static void main(String[] args) {  
 try (FileWriter writer = new FileWriter("test.txt")) {  
 writer.write("Hello, World!");  
 System.out.println("Data written to file.");  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

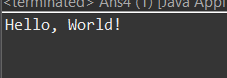
**OUTPUT:**



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

import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
  
public class Ans4 {  
 public static void main(String[] args) {  
 try (BufferedReader br = new BufferedReader(new FileReader("test.txt"))) {  
 String line;  
 while ((line = br.readLine()) != null) {  
 System.out.println(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

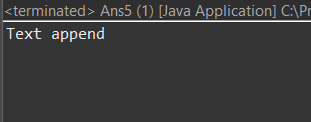
**OUTPUT:**



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

import java.io.FileWriter;  
import java.io.IOException;  
  
public class Ans5 {  
 public static void main(String[] args) {  
 try (FileWriter writer = new FileWriter("test.txt", true)) {  
 writer.write("fygjuyogr”);  
 System.out.println("Text appended.");  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

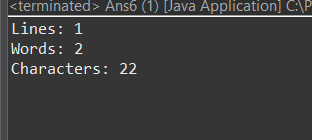
**OUTPUT:**



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

import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
  
public class Ans6 {  
 public static void main(String[] args) {  
 int lines = 0, words = 0, chars = 0;  
 try (BufferedReader br = new BufferedReader(new FileReader("test.txt"))) {  
 String line;  
 while ((line = br.readLine()) != null) {  
 lines++;  
 words += line.split("\s+").length;  
 chars += line.length();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 System.out.println("Lines: " + lines);  
 System.out.println("Words: " + words);  
 System.out.println("Characters: " + chars);  
 }  
}

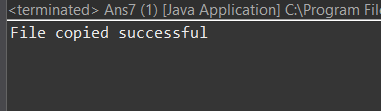
**OUTPUT:**



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

import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
  
public class Ans7 {  
 public static void main(String[] args) {  
 try (FileReader fr = new FileReader("test.txt");  
 FileWriter fw = new FileWriter("copy.txt")) {  
 int c;  
 while ((c = fr.read()) != -1) {  
 fw.write(c);  
 }  
 System.out.println("File copied successful");  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

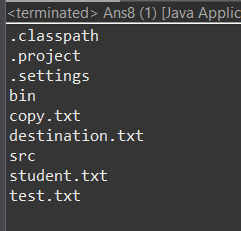
**OUTPUT:**



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

import java.io.File;  
  
public class Ans8 {  
 public static void main(String[] args) {  
 File folder = new File(".");  
 File[] list = folder.listFiles();  
 if (list != null) {  
 for (File f : list) {  
 System.out.println(f.getName());  
 }  
 }  
 }  
}

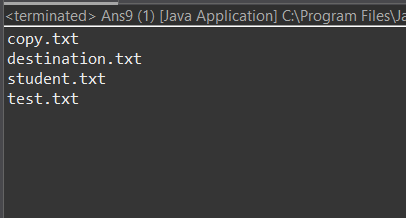
**OUTPUT:**



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

import java.io.File;  
import java.io.FilenameFilter;  
  
public class Ans9 {  
 public static void main(String[] args) {  
 File folder = new File(".");  
 File[] txtFiles = folder.listFiles(new FilenameFilter() {  
 public boolean accept(File dir, String name) {  
 return name.endsWith(".txt");  
 }  
 });  
 for (File f : txtFiles) {  
 System.out.println(f.getName());  
 }  
 }  
}

**OUTPUT:**



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

import java.io.\*;  
  
class Student implements Serializable {  
 String name;  
 int age;  
 Student(String name, int age) { this.name = name; this.age = age; }  
}  
  
public class SerializeStudent {  
 public static void main(String[] args) {  
 Student s = new Student("John", 20);  
 try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("student.ser"))) {  
 oos.writeObject(s);  
 } catch (IOException e) { e.printStackTrace(); }  
 try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("student.ser"))) {  
 Student s2 = (Student) ois.readObject();  
 System.out.println(s2.name + " - " + s2.age);  
 } catch (IOException | ClassNotFoundException e) { e.printStackTrace(); }  
 }  
}

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

import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
public class ScannerReadFile {  
 public static void main(String[] args) throws FileNotFoundException {  
 Scanner sc = new Scanner(new File("test.txt"));  
 while (sc.hasNext()) {  
 System.out.println(sc.next());  
 }  
 sc.close();  
 }  
}

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

import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
public class WordCount {  
 public static void main(String[] args) throws FileNotFoundException {  
 String search = "Hello";  
 int count = 0;  
 Scanner sc = new Scanner(new File("test.txt"));  
 while (sc.hasNext()) {  
 if (sc.next().equals(search)) count++;  
 }  
 sc.close();  
 System.out.println("Occurrences of '" + search + "': " + count);  
 }  
}

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

import java.nio.file.\*;  
  
public class FileOperations {  
 public static void main(String[] args) throws Exception {  
 Path path = Paths.get("temp.txt");  
 Files.createFile(path);  
 Files.move(path, Paths.get("temp\_renamed.txt"), StandardCopyOption.REPLACE\_EXISTING);  
 Files.delete(Paths.get("temp\_renamed.txt"));  
 System.out.println("File created, moved, and deleted successfully.");  
 }  
}

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

import java.nio.file.\*;  
import java.io.IOException;  
import java.util.List;  
  
public class ReadAllLinesExample {  
 public static void main(String[] args) throws IOException {  
 List<String> lines = Files.readAllLines(Paths.get("test.txt"));  
 for (String line : lines) {  
 System.out.println(line);  
 }  
 }  
}

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

import java.nio.file.\*;  
import java.io.IOException;  
  
public class WalkDirectory {  
 public static void main(String[] args) throws IOException {  
 Files.walk(Paths.get("."))  
 .filter(Files::isRegularFile)  
 .forEach(System.out::println);  
 }  
}

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

import java.nio.file.\*;  
import java.io.IOException;  
  
public class CopyFileNIO {  
 public static void main(String[] args) throws IOException {  
 Files.copy(Paths.get("test.txt"), Paths.get("copy.txt"), StandardCopyOption.REPLACE\_EXISTING);  
 System.out.println("File copied successfully using Files.copy().");  
 }  
}

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

import java.nio.file.\*;  
import java.io.IOException;  
  
public class FileSize {  
 public static void main(String[] args) throws IOException {  
 long size = Files.size(Paths.get("test.txt"));  
 System.out.println("Size: " + size + " bytes");  
 }  
}

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

import java.io.\*;  
  
class Employee implements Serializable {  
 String name;  
 int id;  
 Employee(String name, int id) { this.name = name; this.id = id; }  
}  
  
public class SerializeEmployee {  
 public static void main(String[] args) {  
 Employee e = new Employee("Alice", 101);  
 try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("employee.ser"))) {  
 oos.writeObject(e);  
 System.out.println("Employee serialized successfully.");  
 } catch (IOException ex) { ex.printStackTrace(); }  
 }  
}

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

import java.io.\*;  
  
public class DeserializeEmployee {  
 public static void main(String[] args) {  
 try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("employee.ser"))) {  
 Employee e = (Employee) ois.readObject();  
 System.out.println("Employee Name: " + e.name);  
 System.out.println("Employee ID: " + e.id);  
 } catch (IOException | ClassNotFoundException ex) { ex.printStackTrace(); }  
 }  
}  
  
class Employee implements Serializable {  
 String name; int id;  
}