```java
//File Handling Practice Programs
//Prepared By : Sumathi R
//https://www.linkedin.com/in/sumathi1989/



package com.practice.java8;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.RandomAccessFile;
import java.io.Reader;
import java.nio.file.Paths;
import java.util.Map.Entry;
import java.util.Properties;
import java.util.Scanner;
import java.util.concurrent.Delayed;

public class FileHandling {

    public static void main(String[] args) {

        FileHandling fh = new FileHandling();
        fh.createFolder("D:\\FileHandlingDemo\\MyDocs");
        fh.createSubFolder("D:\\FileHandlingDemo\\MyDocs\\Reports");
        fh.createSubFoldersinOneGo();
        fh.createNewFile();
        fh.writingContent();
        fh.createDynamicFile();
        File basefolder = new File("D:\\FileHandlingDemo\\MyDocs");
        System.out.println("Scanning folder :: " + basefolder.getAbsolutePath());
        fh.scanFiles(basefolder);
        fh.listingTextFiles(basefolder);
        fh.performRenameDelete();
        fh.performingRenameDelete();
        fh.fileReadingWritingDemo();
        fh.fileReadingWritingDemoWithCharArray();
        fh.copyFileWithLineNumbers();
        fh.appendAndCountSentencesInFile();
        fh.copyImageFile();
        fh.searchWordInFile();
        fh.propertiesDemo();
        fh.mergeFiles();
        fh.splitFile();
        fh.randomAccessDemo();

    }
```

```java

    /*1.Write a program that checks if a folder named MyDocs exists in your project directory.
    If it does not exist, create it.
    Print whether the folder was newly created or already existed.
     */

    public void createFolder(String folderpath) {

        File file = new File(folderpath);
        if(!file.exists()) {
            System.out.println("Folder does not exist..Creating new Folder...");
            boolean created = file.mkdir();
            System.out.println("Folder created :: " + created);
        }
        else {
            System.out.println("Folder already exist!");
        }
    }


    /*2.Inside MyDocs, create a subfolder Reports.
     Ensure the program creates both folders if they don't exist.
     Print the full absolute path of the Reports folder.*/

    public void createSubFolder(String folderpath) {

        File file = new File(folderpath);
        if (!file.exists()) {
            System.out.println("Folder/Sub folder missing...creating new folder...");
            boolean created = file.mkdirs();
            if (created) {
                System.out.println("Folder/Subfolder successfully created.");
            } else {
                System.out.println("Failed to create Folder/Subfolder.");
            }
        } else {
            System.out.println("Folder/Subfolder already exists! ");
        }
        System.out.println("Full path :: " + file.getAbsolutePath());
    }



    /*3. Create a folder structure like this in one execution
      MyDocs
        └── 2025
            ├── Jan
            ├── Feb
            └── Mar    */
    public void createSubFoldersinOneGo() {
        File base = new File("D:\\FileHandlingDemo\\MyDocs\\2025");
        if(!base.exists()) {
            boolean created = base.mkdirs();
            System.out.println("Base folder created: " + created);
        }
        String[] months = {"Jan" , "Feb" , "Mar"};
        for(String month : months) {
            File sub = new File(base, month);
            if(!sub.exists()) {
                boolean created = sub.mkdir();
                System.out.println("Created " + sub.getAbsolutePath() + " :: " + created
                );
            }
            else {
                System.out.println(sub.getAbsolutePath() + " already exists.");
            }
        }
    }
```

```java


    /*4. Inside MyDocs, create a text file named notes.txt.
     If it already exists, print a message saying so.
     Otherwise, create it and write "Hello File Handling!" inside.
     */

     public void createNewFile() {

         File file = new File("D:\\FileHandlingDemo\\MyDocs\\note.txt");
         if (!file.exists()) {
             System.out.println("File doesn't exist..Creating a new file!");
             try {
                 boolean created = file.createNewFile();
                 if (created)
                     System.out.println("File created successfully!");

                 try (FileWriter writer = new FileWriter(file)) {
                     writer.write("Hello file handling!");
                     System.out.println("Content written successfully.");
                 }
             } catch (IOException e) {
                 e.printStackTrace();
             }
         } else {
             System.out.println("File already exists!");
         }
     }



     /*5. In the folder MyDocs/Reports, create three files:
     report1.txt , report2.txt, summary.txt
     Write some dummy content in each.
      */

     public void writingContent() {

         File base = new File("D:\\FileHandlingDemo\\MyDocs\\Reports");
         if(!base.exists()) {
             System.out.println("Folder doen't exist..Creating new folder...");
             boolean craeted = base.mkdirs();
             System.out.println("Base folder :: " + craeted);
         }
         else {
             System.out.println("Base foldr alreday exists :: " + base.getAbsolutePath());
         }
         String[] files = {"report1.txt" , "report2.txt" , "summary.txt"};
         for(String filename : files) {
              File sub = new File(base, filename);
              if(!sub.exists()) {
                  try {
                     boolean created =sub.createNewFile();
                     if(created) {
                         try(FileWriter writer = new FileWriter(sub)){
                              writer.write("This is dummy content for " + sub.getName());
                              System.out.println("Content written to " + sub.getName());
                         }
                     }
                  } catch (IOException e) {
                     e.printStackTrace();
                  }
              }
              else {
                  System.out.println("File already exists: " + sub.getName());
              }
         }
     }

```

```java


    /*6. Dynamic File Creation - Ask the user for a folder name (e.g., Projects).
    Create that folder inside MyDocs.
    Inside it, create README.txt with the content:This is the <folderName> project
    folder.
    */

    public void createDynamicFile() {

        Scanner sc = new Scanner(System.in);
        System.out.println("Hey User, Enter the folder name : ");
        String folderName = sc.nextLine();

        File folder = new File("D:\\FileHandlingDemo\\MyDocs\\" + folderName);

        if (!folder.exists()) {
            System.out.println("Folder doesn't exist! Creating new subfolder...");
            boolean created = folder.mkdirs();
            System.out.println("Folder created successfully!" + created);
        } else {
            System.out.println("Folder already exists!");
        }
        File readme = new File(folder, "README.txt");
        try (FileWriter writer = new FileWriter(readme)) {
            writer.write("This is the " + folderName + " project folder");
            System.out.println("README.txt created with content!");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }



    /*7. Check & Report
    Write a program that scans MyDocs and lists:
    All subfolders
    All files (with full path)*/

    public void scanFiles(File base) {

        if (!base.exists()) {
            System.out.println("Folder not found!");
            return;
        }
        File[] fileFolders = base.listFiles();
        if (fileFolders == null || fileFolders.length == 0) {
            System.out.println("No Folders/Files found!");
        }
        for (File f : fileFolders) {
            if (f.isDirectory()) {
                System.out.println("SubFolder :: " + f.getAbsolutePath());
                scanFiles(f);
            } else if (f.isFile())
                System.out.println("File :: " + f.getAbsolutePath());
        }

    }


    /*8. Writing a program for Listing only text files and
     * listing files which is greater than >100KB */

    public void listingTextFiles(File basefolder) {
        if (!basefolder.exists()) {
            System.out.println("No Folders/Files found! ");
        }

        File[] fa = basefolder.listFiles();
```

```
274
275             for (File f : fa) {
276                 if (f.isFile()) {
277                     String fname = f.getName();
278                     /*int pos = fname.lastIndexOf(".");
279                     String str = fname.substring((pos + 1), fname.length());
280                     if (str.equals("txt")) {
281                         System.out.println("1.." + fname);
282                     }*/
283                     if(fname.toLowerCase().endsWith(".txt")) {
284                         System.out.println("1.." + fname);
285                     }
286                     if (f.length() > 100000 ) {
287                         System.out.println("Large File (>100KB) :: " + fname + " / size : " +
                             f.length() + " bytes");
288                     }
289                 }
290             }
291         }
292
293
294
295         /* 9. Write a Java program that:
296         Creates a file named draft.txt inside D:\FileHandlingDemo\MyDocs.
297         Renames this file to final.txt.Deletes the renamed file.
298         At each step (create, rename, delete), print whether the operation was successful or
        failed. */
299
300         public void performRenameDelete() {
301
302             File draftedFile = new File("D:\\FileHandlingDemo\\MyDocs\\draft.txt");
303             if (draftedFile.exists()) {
304                 boolean deleted = draftedFile.delete();
305                 System.out.println("Old draft.txt file deleted :: " + deleted);
306             }
307             try {
308                 boolean created = draftedFile.createNewFile();
309                 System.out.println("New File created successfully " + created);
310                 File finalFile = new File("D:\\FileHandlingDemo\\MyDocs\\final.txt");
311                 boolean renamed = draftedFile.renameTo(finalFile);
312                 System.out.println("File is renamed successfully :: " + renamed);
313
314                 if(finalFile.exists()) {
315                     boolean deleted = finalFile.delete();
316                     System.out.println("Renamed file is deleted successfully :: " + deleted);
317                 }
318             } catch (IOException e) {
319                 e.printStackTrace();
320             }
321
322         }
323
324
325
326         /*10.Write a Java program that:
327         Creates a folder named TempData inside D:\FileHandlingDemo\MyDocs.
328         Renames this folder to ArchiveData.
329         Asks the user: “Do you really want to delete ArchiveData? (yes/no)”
330         If the user answers yes and the folder is empty → delete it.
331         If the folder is not empty, display a warning: “Folder is not empty, cannot
        delete!”*/
332
333         public void performingRenameDelete() {
334
335             File tempFolder = new File("D:\\FileHandlingDemo\\MyDocs\\tempFolder");
336             if (tempFolder.exists()) {
337                 boolean oldFolder = tempFolder.delete();
338                 System.out.println("Old tempFolder is deleted : " + oldFolder);
339             }
```

```
340         boolean created = tempFolder.mkdir();
341         System.out.println("New tempFolder created : " + created);
342 
343         File archiveFolder = new File("D:\\FileHandlingDemo\\MyDocs\\ArchiveData");
344         boolean renamed = tempFolder.renameTo(archiveFolder);
345         System.out.println("Folder is renamed : " + renamed);
346 
347         System.out.println("Do you want to delete ArchiveData? (yes/no)");
348         Scanner input = new Scanner(System.in);
349         String yesOrNo = input.nextLine();
350 
351         if (yesOrNo.equalsIgnoreCase("yes")) {
352             {
353                 if (archiveFolder.isDirectory() && archiveFolder.list().length == 0) {
354                     boolean deleted = archiveFolder.delete();
355                     System.out.println("Archive folder is deleted : " + deleted);
356                 } else {
357                     System.out.println("Folder is not empty, cannot delete!");
358                 }
359             }
360         } else {
361             System.out.println("Archive folder is not deleted!");
362         }
363     }
364 
365 
366 
367     /* 11. Write a Java program that:
368     Creates a file named greeting.txt inside D:\FileHandlingDemo\MyDocs.
369     Writes the text "Hi Sumathi, how are you?" into this file using FileWriter.
370     Reads the content back from the file using FileReader.
371     Prints the content on the console.
372     Counts the number of characters in the file (excluding spaces and punctuation) and
    prints the count.*/
373 
374     public void fileReadingWritingDemo() {
375 
376         File file = new File("D:\\FileHandlingDemo\\MyDocs\\greeting.txt");
377         if (file.exists()) {
378             boolean deleted = file.delete();
379             System.out.println("old file is deleted : " + deleted);
380         }
381 
382         try {
383             boolean created = file.createNewFile();
384             System.out.println("new file is created : " + created);
385 
386             FileWriter writer = new FileWriter(file);
387             writer.write("Hi Sumathi, How are you?");
388             writer.flush();
389             writer.close();
390 
391             FileReader reader = new FileReader(file);
392             int output = reader.read();
393             int charCount = 0;
394             while (output != -1) {
395                 char c = (char) output;
396                 System.out.print(c);
397 
398                 if(Character.isLetterOrDigit(c)) {
399                     charCount++;
400                 }
401                 output = reader.read();
402             }
403             reader.close();
404             System.out.println("\n Character count is(excluding spaces and punctuation)
                : " + charCount);
405         }
406 
```

```
        catch (IOException e) {
            e.printStackTrace();
        }

    }



    /*12. Create a file named story.txt inside D:\FileHandlingDemo\MyDocs.
    Write the following content into it:
    Once upon a time, there was a developer named Sumathi.
    She loved learning Java file handling step by step.
    Now, using FileReader with a char array buffer:
    Read the file content.
    Print the content line by line.
    Count and display:
    Total number of characters
    Total number of words      */

    public void fileReadingWritingDemoWithCharArray() {

        File file = new File("D:\\FileHandlingDemo\\MyDocs\\story.txt");
        if (file.exists()) {
            boolean deleted = file.delete();
            System.out.println("Old fils is deleted :: " + deleted);
        }
        try {
            boolean created = file.createNewFile();
            System.out.println("File is created :: " + created);

            FileWriter writer = new FileWriter(file);
            writer.write("Once upon a time, there was a developer named Sumathi.  \r\n"
                    + " She loved learning Java file handling step by step");
            writer.flush();
            writer.close();

            FileReader reader = new FileReader(file);
            /*
             * // line by line reading //using previous reader method
             * int output = reader.read();
             * int charCOunt = 0;
             * while (output != -1) {
             * charCOunt++;
             * System.out.print((char) output);
             * output =reader.read(); }
             * System.out.println("\nCharacter count is :: " + charCOunt);
             * reader.close();
             */

            char[] ch = new char[(int) file.length()];
            System.out.println(ch.length);
            reader.read(ch);
            for(char ch1 : ch) {
                System.out.println(ch1);
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

```java

    /*13. Create a file named notes.txt inside D:\FileHandlingDemo\MyDocs.
    Write 3–4 lines of text into it (e.g., some study notes).
    Read the file line by line using BufferedReader.
    While reading, write the same lines into a new file named notes_copy.txt,
    but add line numbers in front of each line.*/

    public void copyFileWithLineNumbers() {

        File file = new File("D:\\FileHandlingDemo\\MyDocs\\notes.txt");
        File copiedFile = new File("D:\\FileHandlingDemo\\MyDocs\\notes_copy.txt");

        if (file.exists()) {
            boolean deleted = file.delete();
            System.out.println("Old file is deleted :: " + deleted);
        }
        if (copiedFile.exists()) {
            boolean deleted = copiedFile.delete();
            System.out.println("Old file of copied file is deleted :: " + deleted);
        }

        try {
            boolean created = file.createNewFile();
            System.out.println("File is created :: " + created);

            boolean copyCreated = copiedFile.createNewFile();
            System.out.println("Copied file is created :: " + copyCreated);

            FileWriter fWriter = new FileWriter(file);
            BufferedWriter bWriter = new BufferedWriter(fWriter);

            FileWriter fWriter2 = new FileWriter(copiedFile);
            BufferedWriter bWriter2 = new BufferedWriter(fWriter2);

            bWriter.write("Tamil");
            bWriter.newLine();
            bWriter.write("English");
            bWriter.newLine();
            bWriter.write("Mathematics");
            bWriter.flush();
            bWriter.close();

            FileReader fReader = new FileReader(file);
            BufferedReader bReader = new BufferedReader(fReader);

            String line = bReader.readLine();
            int lineCount = 0;
            while (line != null) {
                lineCount++;
                bWriter2.write("" + lineCount + "." + line);
                bWriter2.newLine();
                System.out.println(line);
                line = bReader.readLine();
            }
            bWriter2.flush();
            bWriter2.close();
            bReader.close();

            System.out.println("File copied successfully with line numbers. Total lines
            = " + lineCount);

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

```java
    /*14. Create a file named journal.txt inside D:\FileHandlingDemo\MyDocs.
    Write some initial content into the file using FileWriter (e.g., "Today is a good day. I practiced Java file handling.").
    Now, manually edit the file and add more sentences.
    Read the entire file back using BufferedReader.
    While reading, print all file content.
    Also, count how many sentences , words, characters are present in the file (hint: split by ., ?, or !)
    Print the total number of sentences, words, char at the end.*/


    public void appendAndCountSentencesInFile() {

        File file = new File("D:\\FileHandlingDemo\\MyDocs\\journal.txt");

        try {
            file.createNewFile();
            BufferedWriter bWriter = new BufferedWriter(new FileWriter(file, true));
            bWriter.write("Today is a good day. I practiced Java file handling.");
            bWriter.newLine();
            bWriter.flush();
            bWriter.close();

            FileReader fReader = new FileReader(file);
            BufferedReader bReader = new BufferedReader(fReader);
            String line = bReader.readLine();
            int lineCount = 0;
            int sentenceCount = 0;
            int wordCount = 0;
            int charCount = 0;
            while (line != null) {
                String[] sentence = line.trim().split("[.?!]");
                sentenceCount = sentenceCount + sentence.length;

                String[] word = line.split(" ");
                wordCount = wordCount + word.length;

                System.out.println(line);

                charCount = charCount + line.length();

                lineCount++;
                line = bReader.readLine();
            }

            bReader.close();
            System.out.println("No of lines " + lineCount);
            System.out.println("No of sentences " + sentenceCount);
            System.out.println("No of words " + wordCount);
            System.out.println("No of characters " + charCount);

        }

        catch (IOException e) {
            e.printStackTrace();
        }

    }
```

```java
    /*15. Write a Java program that reads an image file (e.g., photo.jpg) from a given location and
     * creates an exact copy of it in the same folder with a new name (e.g., photo_copy.jpg). */

    public void copyImageFile() {

        try {
            InputStream input = new FileInputStream("D:\\My Life.png");
            OutputStream output = new FileOutputStream("D:\\Always.png");
            int content = input.read();
            while (content != -1) {
                output.write(content);
                content = input.read();
            }
            output.flush();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }


    /* 16. Write a Java program that:
    Reads a .txt file (e.g., story.txt) line by line.
    Asks the user to input a word to search.
    Searches for the word in each line (case-insensitive).
    Prints the line numbers where the word is found.
    If the word is not found in the entire file → print "Word not found!".*/

    public void searchWordInFile() {

        File file = new File("D:\\FileHandlingDemo\\MyDocs\\story.txt");

        if (!file.exists()) {
            System.out.println("Folder not found!");
            return;
        }
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the word to search!");
        String input = sc.nextLine().toLowerCase();
        sc.close();

        try {
            FileReader fReader = new FileReader(file);
            BufferedReader bReader = new BufferedReader(fReader);

            String line = bReader.readLine();
            int lineNo = 1;
            int wordCount = 0;
            boolean found = false;
            while (line != null) {

                String[] word = line.toLowerCase().split(" ");
                for (String w : word) {
                    if (w.contains(input)) {
                        System.out.println("The word " + input + " found at the " +
                        lineNo + ":" + line);
                        found = true;
                        wordCount++;
                    }
                }
                lineNo++;
                line = bReader.readLine();
            }
            lineNo++;
            line = bReader.readLine();
```

```
677                 bReader.close();
678                 fReader.close();
679 
680                 if (!found) {
681                     System.out.println("Word not found in the file!");
682                 } else {
683                     System.out.println("The word '" + input + "' occurred " + wordCount + "
                        times in the file.");
684                 }
685             }
686 
687             catch (FileNotFoundException e) {
688                 e.printStackTrace();
689             } catch (IOException e) {
690                 e.printStackTrace();
691             }
692 
693         }
694 
695 
696 
697         /* 17. Create a program to manage application configuration settings using a
            .properties file.
698         Create a file named config.properties.
699         Add entries like:
700         db.username=admin
701         db.password=12345
702         db.url=jdbc:oracle:thin:@localhost:1521:xe
703         Write a Java program that will:
704         Read values from the config.properties file (print them on console).
705         Update a property value (for example, change the db.password).
706         Save the updated properties back into the same file. */
707 
708         public void propertiesDemo() {
709 
710             File file = new File("D:\\FileHandlingDemo\\MyDocs\\config.properties");
711 
712             Properties prop = new Properties();
713             try {
714                 if (file.exists()) {
715                     FileInputStream fis = new FileInputStream(file);
716                     prop.load(fis);
717                     fis.close();
718                 }
719                 System.out.println("Current properties ::: ");
720 
721                 for (Object keyObj : prop.keySet()) {
722                     String key = (String) keyObj;
723                     String value = prop.getProperty(key);
724                     System.out.println("key" + "=" + key + "  ::  " + "value" + "=" + value);
725                 }
726                 prop.setProperty("db.username", "system");
727                 prop.setProperty("db.password", "234");
728                 prop.setProperty("db.url", "jdbc:oracle.thin:@localhost:1521:orcl");
729 
730                 FileOutputStream fos = new FileOutputStream(file);
731                 prop.store(fos, "DB Configuration");
732                 fos.close();
733 
734                 System.out.println("\n Updated properties saved to file..!");
735             } catch (FileNotFoundException e) {
736                 e.printStackTrace();
737             } catch (IOException e) {
738                 e.printStackTrace();
739             }
740 
741         }
742 
743 
```

```java
    /*18. Merge Files
    You have 3 text files: file1.txt , file2.txt ,file3.txt
    Write a Java program to merge their contents into a single file named merged.txt.
    Make sure each file's content starts on a new line in the merged file.*/

    public void mergeFiles() {

        String[] sourceFiles = { "D:\\FileHandlingDemo\\MyDocs\\file1.txt",
        "D:\\FileHandlingDemo\\MyDocs\\file2.txt",
                "D:\\FileHandlingDemo\\MyDocs\\file3.txt" };

        String mergedFile = "D:\\FileHandlingDemo\\MyDocs\\merged.txt";

        try {
            BufferedWriter writer = new BufferedWriter(new FileWriter(mergedFile));

            for (String src : sourceFiles) {
                FileReader fr = new FileReader(src);
                BufferedReader br = new BufferedReader(fr);

                String line;
                while ((line = br.readLine()) != null) {
                    writer.write(line);
                    writer.newLine();
                }

                writer.newLine();
                br.close();
                fr.close();
            }

            writer.close();
            System.out.println("Files merged successfully into: " + mergedFile);

        } catch (IOException e) {
            e.printStackTrace();
        }
    }



    /*19. Split File
    Take a large text file (say merged.txt).
    Write a Java program to split it into multiple smaller files of N lines each
    (example: split every 5 lines into a new file like part1.txt, part2.txt, …).*/

    public void splitFile() {
        String sourceFile = "D:\\FileHandlingDemo\\MyDocs\\merged.txt";
        int linesPerFile = 5;

        try {
            BufferedReader reader = new BufferedReader(new FileReader(sourceFile));

            String line;
            int fileCount = 1;
            int lineCount = 0;

            BufferedWriter writer = new BufferedWriter(
                    new FileWriter("D:\\FileHandlingDemo\\MyDocs\\split_" + fileCount +
                    ".txt"));

            while ((line = reader.readLine()) != null) {
                writer.write(line);
                writer.newLine();
                lineCount++;

                if (lineCount == linesPerFile) {
                    writer.close();
```

```
                    fileCount++;
                    lineCount = 0;

                    writer = new BufferedWriter(
                            new FileWriter("D:\\FileHandlingDemo\\MyDocs\\split_" +
                            fileCount + ".txt"));
                }
            }

            writer.close();
            reader.close();

            System.out.println("File split into " + fileCount + " parts successfully.");

        } catch (IOException e) {
            e.printStackTrace();
        }
    }



    /*20. Create a program using RandomAccessFile that:
    Writes some text to a file.
    Reads content from the beginning.
    Moves the file pointer to a specific position (say 10th byte) and writes new data.
    Reads the file again to show the updated content.*/

    public void randomAccessDemo() {

        try {
            RandomAccessFile raf = new RandomAccessFile(
            "D:\\FileHandlingDemo\\MyDocs\\random.txt", "rw");

            raf.writeUTF("Hello, This is a RandomAccess Demo by Sumathi!");

            raf.seek(0);
            System.out.println("Reading from Start.....");
            System.out.println(raf.readUTF());

            raf.seek(10);
            raf.writeUTF("inserted text");

            raf.seek(0);
            System.out.println("\nReading after update: ");
            System.out.println(raf.readUTF());

            raf.close();

        } catch (IOException e) {
            e.printStackTrace();
        }

    }

}


    //Prepared By : Sumathi R
    //https://www.linkedin.com/in/sumathi1989/
```