FileHandling.java

```
package com;
import java.util.Scanner;
public class FileHandling {
	public static void main(String args[])  {
		Scanner sc=new Scanner(System.in);
		String fileName ="";
		FileOperations op = new FileOperations();
	 System.out.println("------FILE HANDLING IN JAVA--------");
	 while(true) {
			System.out.println("Choose your Option");
			System.out.println("Enter 'C' for Creat a File");
			System.out.println("Enter 'W' for Write into File");
			System.out.println("Enter 'R' for Read from File");
			System.out.println("Enter 'A' for Append content to  File");
			System.out.println("Enter 'D' for Delete a  File");
			System.out.println("Enter 'N' for Rename a  File");
			System.out.println("Enter 'X' for Quit");
			String option = sc.nextLine().trim();

			switch(option)
			{
			  case "C" :
				System.out.println("Enter New File Name :");
				fileName = sc.nextLine();
				op.createFile(fileName);
				break;
			  case "W" :
				System.out.println("Enter  File Name to write :");
				fileName = sc.nextLine();
				op.writeToFile(fileName);
				break;
			  case "R" :
				System.out.println("Enter File Name for read :");
				fileName = sc.nextLine();
				op.readFromFile(fileName);
				break;
			  case "A" :
				System.out.println("Enter File Name for append content :");
				fileName = sc.nextLine();
				op.appendToFile(fileName);
				break;
			  case "D" :
				System.out.println("Enter File Name for delete :");
				fileName = sc.nextLine();

				op.deleteFile(fileName);
				break;
			  case "N" :
				System.out.println("Enter  Old File Name  :");
				String OldfileName = sc.nextLine();
				System.out.println("Enter  New File Name  :");
				String NewfileName = sc.nextLine();
				op.renameFile(OldfileName,NewfileName);
				break;
			  case "X" :
```

```
                                System.out.println("Thank you for using this Application!!! Application
exit successfully!");
                                System.exit(0);
                                break;
                          default :
                                System.out.println("Invalid file Option choosen ! try again !!! ");
                        }
                        System.out.println("Do you want to continue(Y/N) ?");
                        if(sc.nextLine().equalsIgnoreCase("N"))
                        {
                             System.out.println("Thank you for using this Application!!! Application
closed successfully!");
                            break;
                        }
                }
                sc.close();

        }
}
```

FileActions.java

```
package com;

public interface FileActions {
     public void createFile(String filename);
     public void writeToFile(String filename);
     public void appendToFile(String filename);
     public void readFromFile(String filename);
     public void deleteFile(String filename);
     public void renameFile(String oldfilename,String newfilename);
}
```

FileOperations.java

```
package com;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;

public class FileOperations implements FileActions {

   public void createFile(String filename)
   {
            File myFile = new File(filename);
            try{
            if(!(myFile.exists())){ // checking file exist or not
                myFile.createNewFile(); // Creating new file
                System.out.println("New File created....");
            }else{
```

```
                System.out.println("File already exisit....");
            }
            }catch (IOException e) {
                e.printStackTrace();
            }
    }
    public void writeToFile(String filename)
    {
             String source = "";
             FileWriter fWrite = null;

             BufferedReader bf=new BufferedReader(new InputStreamReader(System.in));

             File myFile = new File(filename);
             try{
                 if(!(myFile.exists())){
                     myFile.createNewFile();
                 }
                 fWrite = new FileWriter(myFile,false);

                 System.out.println("Write 'stop' when you finish writing file ");
                 while(!(source=bf.readLine()).equalsIgnoreCase("stop")){
                      fWrite.write(source + System.getProperty("line.separator"));
              }
                 System.out.println("File write complete.....");

             }catch (IOException e) {
                 e.printStackTrace();
             }finally{
                 if(fWrite != null)
                     try { fWrite.close(); } catch (IOException e) { e.printStackTrace(); }

             }
    }
    public void appendToFile(String filename)
    {
             String source = "";
             FileWriter fWrite = null;

             BufferedReader bf=new BufferedReader(new InputStreamReader(System.in));

             File myFile = new File(filename);
             try{
                 if(!(myFile.exists())){
                     myFile.createNewFile();
                 }
                 fWrite = new FileWriter(myFile,true); // true for appending content to the existing
file

                 System.out.println("Write 'stop' when you finish appending file ");
              while(!(source=bf.readLine()).equalsIgnoreCase("stop")){
               fWrite.append(source+ System.getProperty("line.separator"));
              }
                 System.out.println("File write complete.....");

             }catch (IOException e) {
                 e.printStackTrace();
             }finally{
                 if(fWrite != null)
```

```
                    try { fWrite.close(); } catch (IOException e) { e.printStackTrace(); }

            }
    }
    public void readFromFile(String filename)
    {
           BufferedReader br = null;
        try{
            FileReader myFile  = new FileReader(filename);
            br = new BufferedReader(myFile);
            String line = null;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        }catch (IOException e) {
            e.printStackTrace();
        }finally{
            if(br != null)
                try{ br.close(); }catch(IOException e){e.printStackTrace();}
        }
    }
    public void deleteFile(String filename)
    {
           try{
            File myFile = new File(filename);
            if(myFile.exists()){
                myFile.delete();
                System.out.println("File deleted successfully....");
            }else{
                System.out.println("File NOT Exisit....");
            }
        }catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void renameFile(String oldFileName,String newFileName) {
           File oriFile = new File(oldFileName);
              File newFile = new File(newFileName);
              if(oriFile.exists()){
                  oriFile.renameTo(newFile);
                  System.out.println("File rename completed....");
              }else{
                  System.out.println("Original file not exist for renaming....");
              }
    }
}
```

Project File Structure: