```
package fileHandling;

import java.io.File;

import java.io.FileReader;
import java.io.FileWriter;

import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.util.Scanner;

public class FileHandling {
        public static void main(String args[]) {
                System.out.println(" w for write mode ");    // 'w' for Overwriting
the existing data.
                System.out.println(" r for read mode ");      // 'r' to read the
existing  data.
                System.out.println(" a for append mode ");     // 'a' for adding
data to the existing data.
                Scanner sc = new Scanner(System.in);
                String s = sc.nextLine();
                if (s.equalsIgnoreCase("r")) {
                        new FileReading();
                } else if (s.equalsIgnoreCase("w") || s.equalsIgnoreCase("a")) {
                        writingToFile(s);

                } else {
                        System.out.println("File Handling didnt intiated");
                }

                sc.close();

        }

        public static void writingToFile(String s) {
                Scanner in = null;
                try {
                        String source = "";
                        File f = new File("C:\\eclipseWorkspace\\file1.txt");
// file is created at specified location

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

                        // Case1: writing data by deleting the existing one
                        FileWriter f0 = null;
                        if (s.equalsIgnoreCase("w")) {
                                f0 = new FileWriter(f, false);
                                System.out.println("Content in file may change ");
                                System.out.println("Type 'no' if you are done
typing..");

                                in = new Scanner(System.in);
                                String s1 = in.nextLine();
                                if (s1.equals("no"))
                                        System.exit(0);
                                System.out.println("Write 'stop' if you are done
typing ");
                                f.delete();
                                f.createNewFile();

                                while (!(source =
bf.readLine()).equalsIgnoreCase("stop")) {
                                        f0.write(source +
System.getProperty("line.separator"));

                                }

                                in.close();
                        }


                        // Case2:appending - adding data to the existing text
without effecting the previous data

                        else {
                                f0 = new FileWriter(f, true);
                                System.out.println("Write 'stop' when you finish
appending(adding few extra data to existing data) file ");

                                while (!(source =
bf.readLine()).equalsIgnoreCase("stop")) {
                                        f0.append(source +
System.getProperty("line.separator"));
                                }
                        }
                        f0.close();

                } catch (Exception e) {
                        System.out.println("Error : ");
                        e.printStackTrace();
                }

        }

}

class FileReading {
        public static String str = null;
        public FileReading() {

                try {
                        File f5 = new File("C:\\eclipseWorkspace\\file1.txt");
 // file is read from  specified directory Path .
                        if (!f5.exists())
                                f5.createNewFile();
                        FileReader fl = new FileReader(f5);
                        BufferedReader bf = new BufferedReader(fl);
                        // For reading till end
                        while ((str = bf.readLine()) != null) {
                                System.out.println(str);
                        }
                        fl.close();
                } catch (Exception e) {
                        System.out.println("Error : ");
                        e.printStackTrace();
                }
        }
}
```