Full Name	Karl Joseph Chetcuti
Email	Karlcht@hotmail.com
Batch	December 2021 Cohort
Project Title	Phase 1 - Implement OOPS using JAVA with Data Structures and Beyond
Git Hub Link:	https://github.com/karlcht/JavaFSPhase1Project.git
Project Submission Date	27th January 2022

Source Code

```
package lockedMe.comPackage;

import java.io.File;
import java.io.FileWriter;
import java.util.LinkedList;
import java.util.Scanner;

public class LockedMe
{

	static final String projectFilesPath = "C:\\Users\\Karl Chetcuti\\Simplilearn Assignments\\SimpliLearn Phase 1 Project - 22-01-2022\\LockedMeFiles";
	static final String errorMsg = "Some error occured, please contact administrator on : karlcht@hotmail.com";
	public static void displayMenu()
	{

		System.out.println("************************************************************");
		System.out.println("\t\t Welcome to LockedMe.com ");
		System.out.println("************************************************************");
		System.out.println("\t Developed by Karl Joseph Chetcuti");
		System.out.println("************************************************************");
		System.out.println("\t\t1. Display all the files");
		System.out.println("\t\t2. Add a new file");
		System.out.println("\t\t3. Delete a file");
		System.out.println("\t\t4. Search a file");
		System.out.println("\t\t5. Exit");
		System.out.println("");
	}

	/**
	 * This function will return all the files from the project directory.
	 */
	public static  void getAllFiles()
```

```
{

        Scanner obj = new Scanner(System.in);
        try
        {


                File directoryPath = new File(projectFilesPath);
            File filesList[] = directoryPath.listFiles();

            if(filesList.length == 0)
            System.out.println("No files in the specified directory:");
            else
                {
                  for(var file : filesList)
               System.out.println(file.getName());
                }

        }

        catch(Exception Ex)
        {

                System.out.println(errorMsg);
        }

        finally
        {

           System.out.println("\n"+"Press return key to continue");
                obj.nextLine();
        }

}
```

```
/**
 * Method to create files but does not permit overwriting of existing files
 */
public static void createFiles()
{
        String fileName;
        int counter;
        Scanner obj = new Scanner(System.in);


        try
        {

                System.out.println("Enter file name");
                fileName = obj.nextLine();


                File f= new File(projectFilesPath+"\\"+fileName);        //file to be delete
                if(f.exists())
                        System.out.println("File Exists do not overwrite !");

                else
                {
                        FileWriter myWriter = new FileWriter(projectFilesPath+"\\"+fileName);
                    System.out.println("Enter how many lines to add");
                    counter = Integer.parseInt(obj.nextLine());
                    for(int i=1; i <= counter; i++)
                    {
                           System.out.println("Enter text for line "+i);
                        myWriter.write(obj.nextLine()+"\n");
                        System.out.println("Successfully wrote to line "+i +"\n");
                    }
```

```
                    System.out.println("Successfully wrote to file ! ");
                    myWriter.close();
                }




        }

        catch (Exception Ex)

                {
            System.out.println(errorMsg);

           }

        finally
        {

           System.out.println("Press return key to continue");
                obj.nextLine();
        }


}

/**
 * This method will delete the files.
 */
public static void deleteFiles()
{

        Scanner obj = new Scanner(System.in);
```

```
        try

        {
                String fileName;

                System.out.println("Enter file name to be deleted.");
                fileName = obj.nextLine();
                File f= new File(projectFilesPath+"\\"+fileName);        //file to be delete

                if(f.exists())
                {
                        f.delete();
                        System.out.println("File: " +fileName+" deleted successfuly");
                }
                else
                System.out.println("File not found");

        }

        catch(Exception Ex)
        {

                System.out.println(errorMsg);
        }

        finally
        {

                System.out.println("Press return key to continue");
                obj.nextLine();
        }
}
```

```
/**
 * This function will search for files in the directory.
 */
public static void searchFiles()
{

        Scanner obj = new Scanner(System.in);

        try

        {



                String fileName;

                System.out.println("Enter file name to be searched.");
                fileName = obj.nextLine();

                File directoryPath = new File(projectFilesPath);
            File listOfFiles[] = directoryPath.listFiles();

            LinkedList <String> filenames = new LinkedList <String>();

            for(var l:listOfFiles)
                    filenames.add(l.getName());
            if(filenames.contains(fileName))
                    System.out.println("File is available.");
            else
                    System.out.println("File not found.");



            }
```

```
            catch(Exception Ex)
            {

                  System.out.println(errorMsg);
            }

            finally
            {

                  System.out.println("Press return key to continue");
                  obj.nextLine();
            }


}

public static void main(String[] args)

{


            int  input;
            Scanner scan = new Scanner(System.in);


            boolean isSuccessful = false;
            while(!isSuccessful)
            {

                  try

                  {
```

```
do
{

        displayMenu();


        System.out.println("Enter your choice");
        input =  Integer.parseInt(scan.nextLine());


        switch(input)
        {
                case 1:
                        getAllFiles();
                        break;
                case 2:
                        createFiles();
                        break;
                case 3:
                        deleteFiles();
                        break;
                case 4:
                        searchFiles();
                        break;
                case 5:
                        System.exit(0);
                        break;
                default:
                        System.out.println("Invalid option");
                        break;
```

```
                                        }

                                }



                                while(input != 0);
                                scan.close();
                                isSuccessful = true;

                        }

                        catch( NumberFormatException Ex)

                        {

                                System.out.println("Please enter a number form 1 to 5.");
                                System.out.println("Press return key to see display menu.");
                                scan.nextLine();

                        }

                        catch( Exception EX)
                        {
                                System.out.println(errorMsg);
                        }

                }


        }
}
```

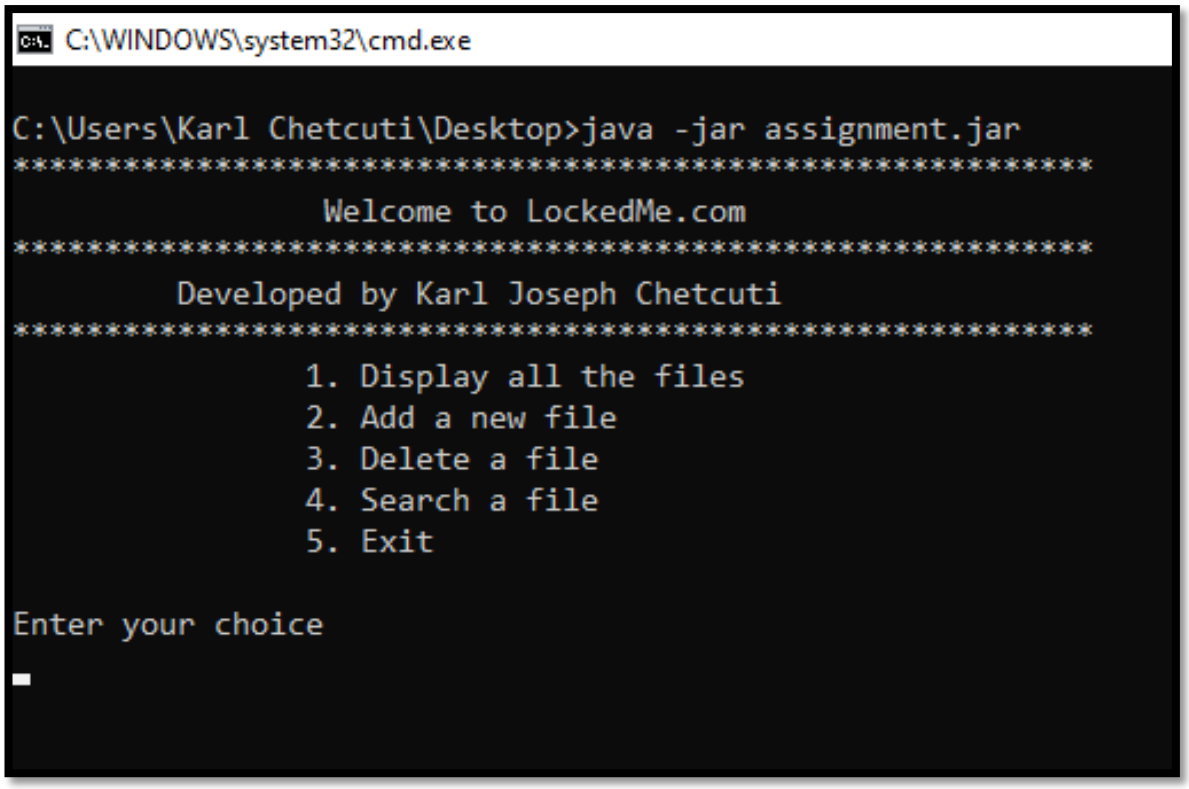


Figure 1 Main Display menu can be seen above.

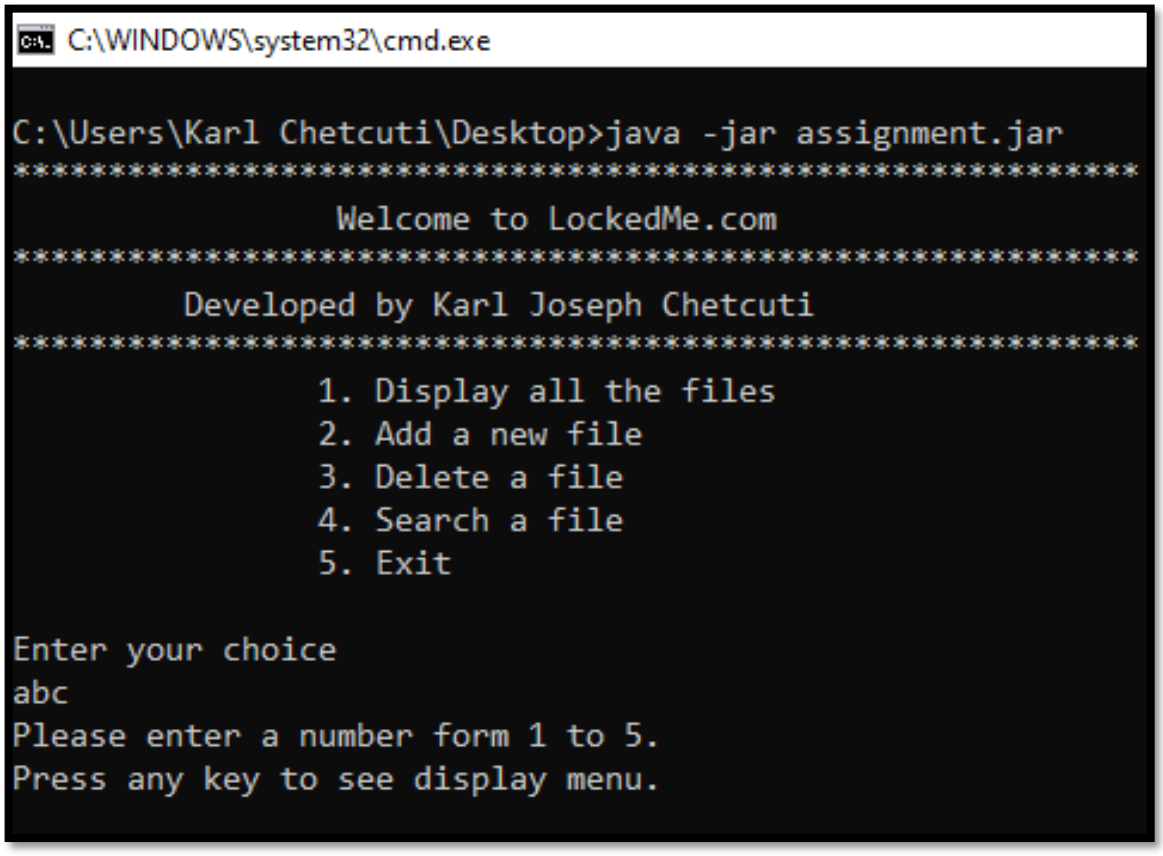


Figure 2 If the input is not an integer error handling will issue an error "Please enter a number from 1 to 5" and the program will loop unless integer 5 is inputted.

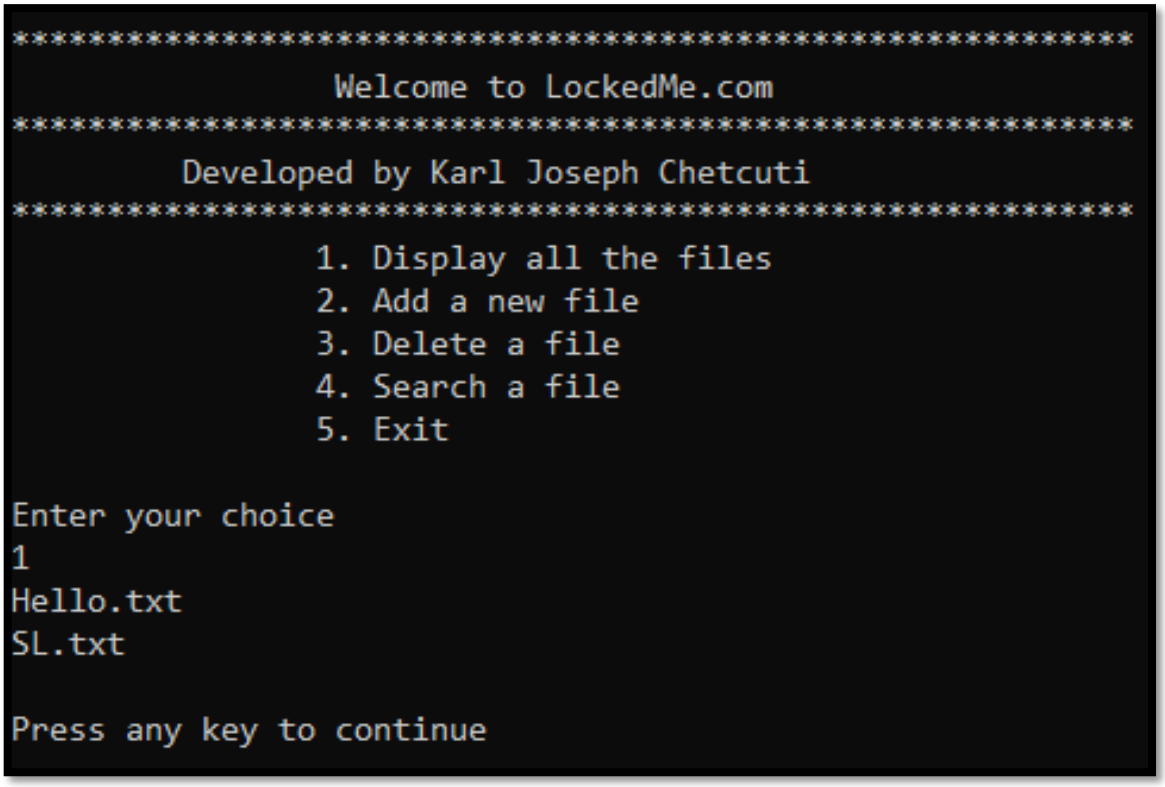


Figure 3 When inputting 1 all files in the folder path will be shown.

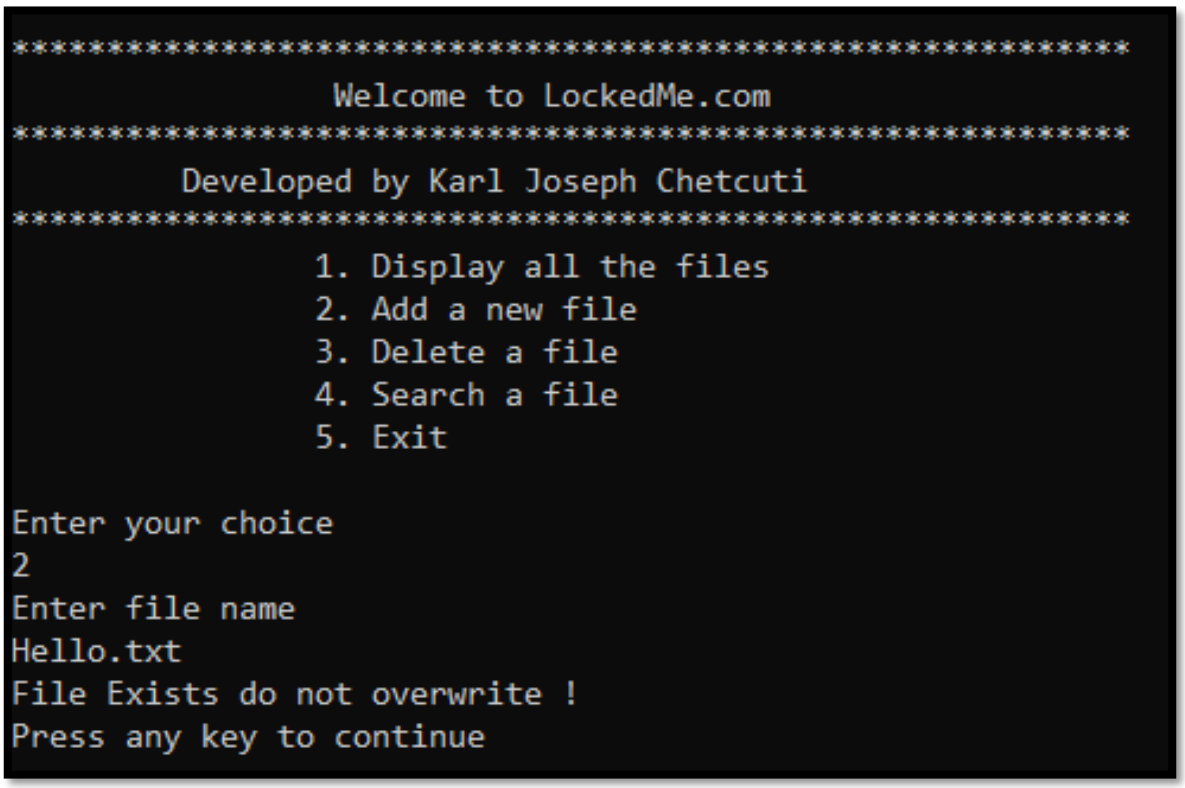


Figure 4 Program will not allow overwriting of file when inputting 2.

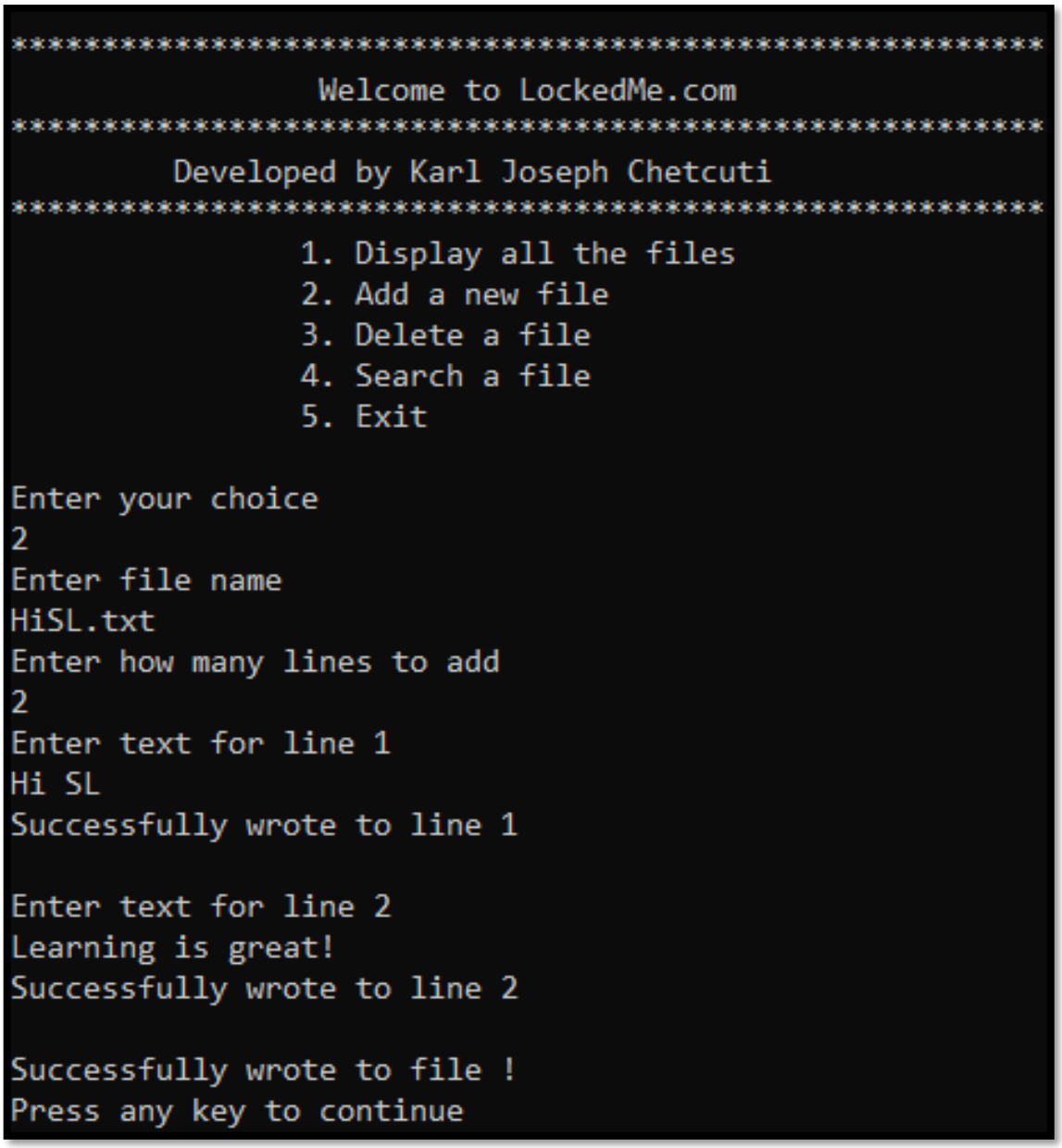


Figure 5 When inputting 2, a new file will be created which will not have a duplicate name.

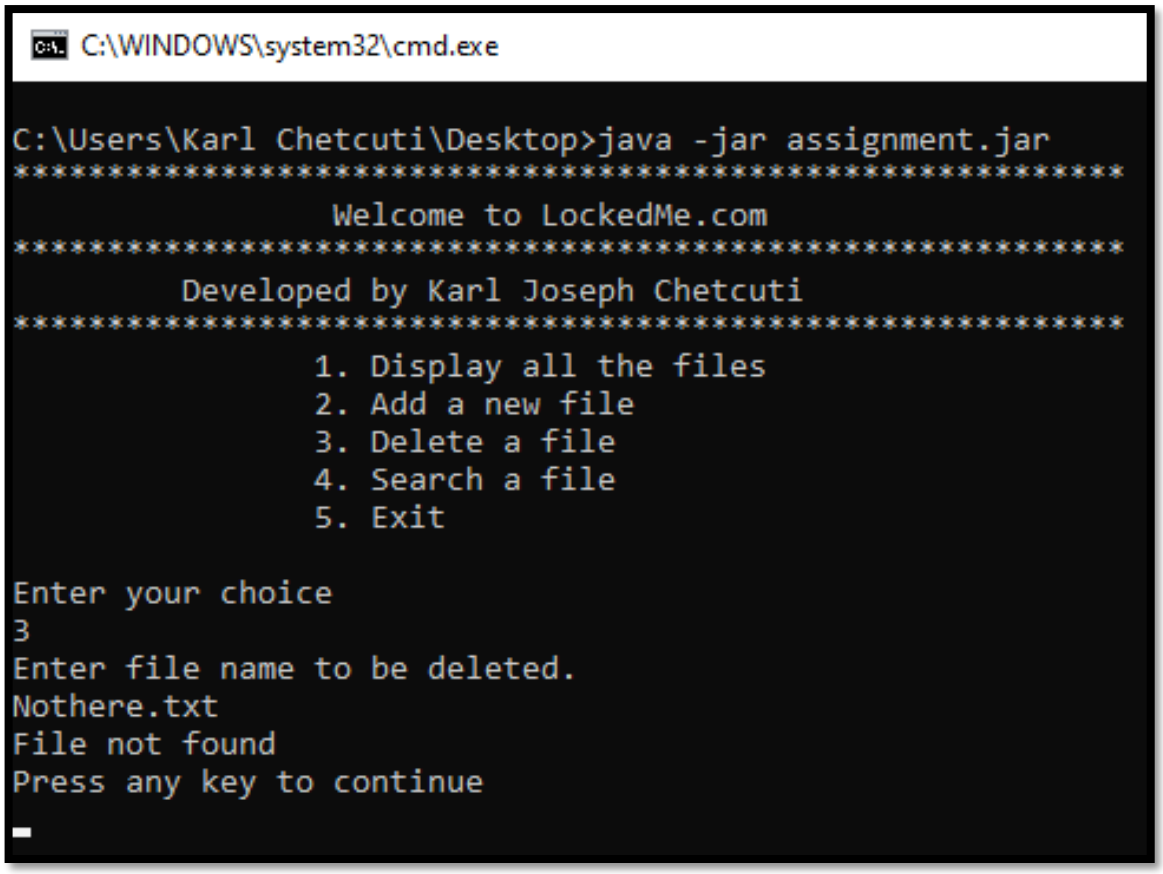


Figure 6 When inputting 3 only an existing file can be deleted. Else program will not allow deletion.

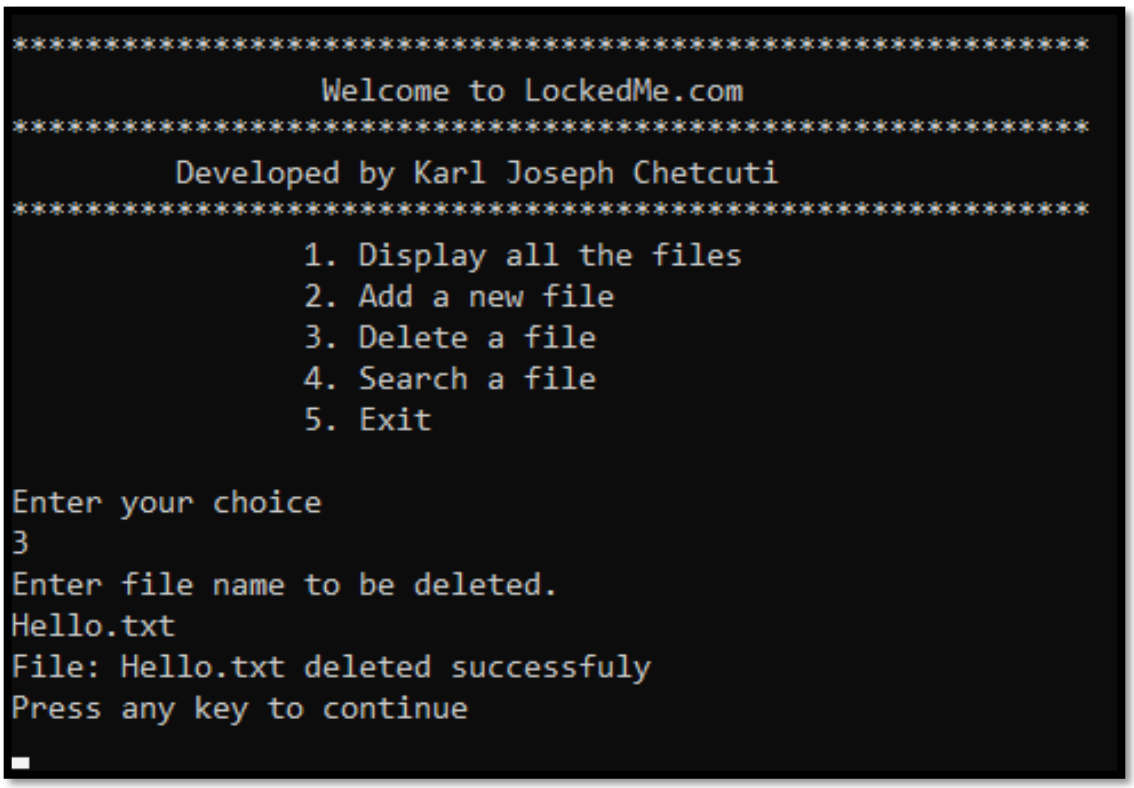


Figure 7 An existing file has been successfully deleted when inputting 3.

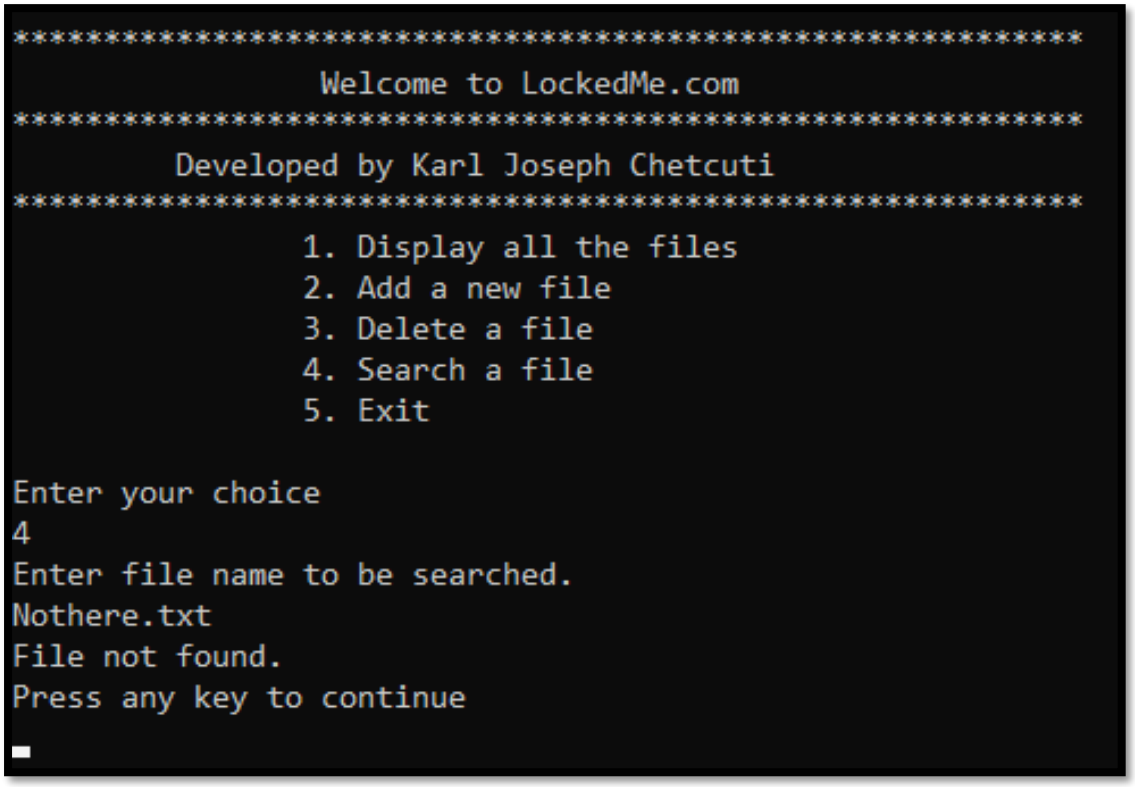


Figure 8 When inputting 4 an existing file name must be searched , else "File not found.".

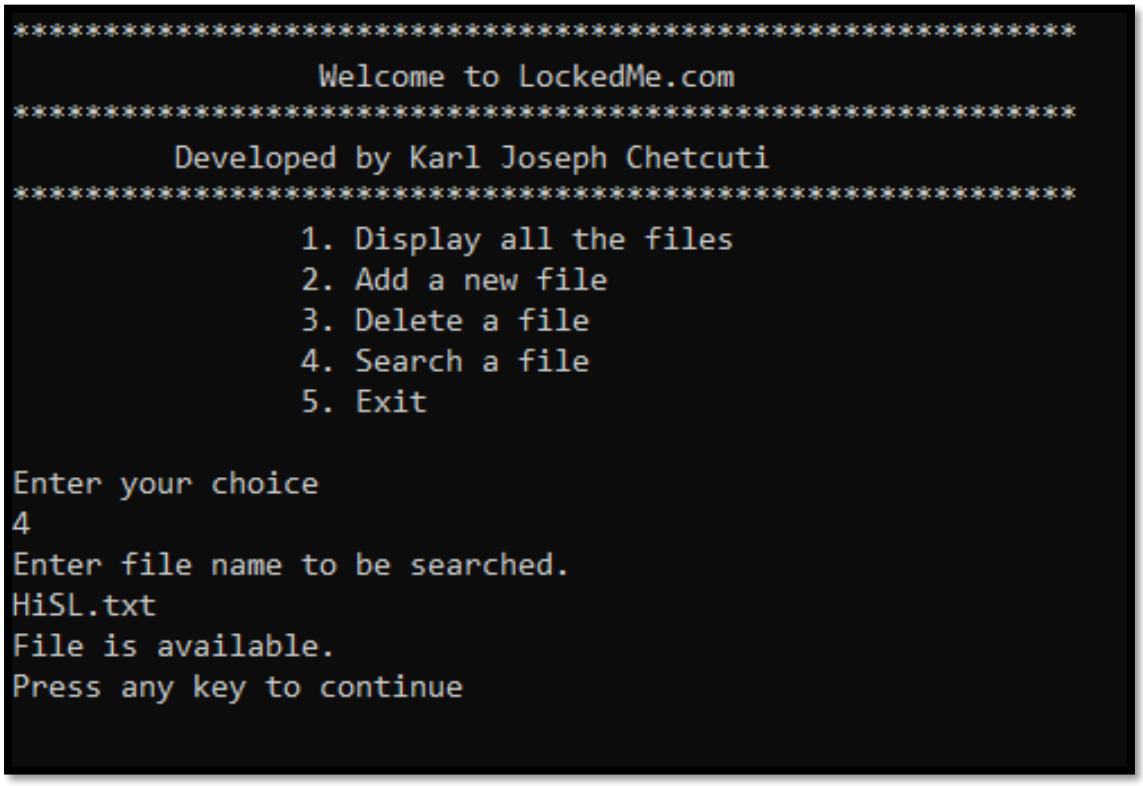
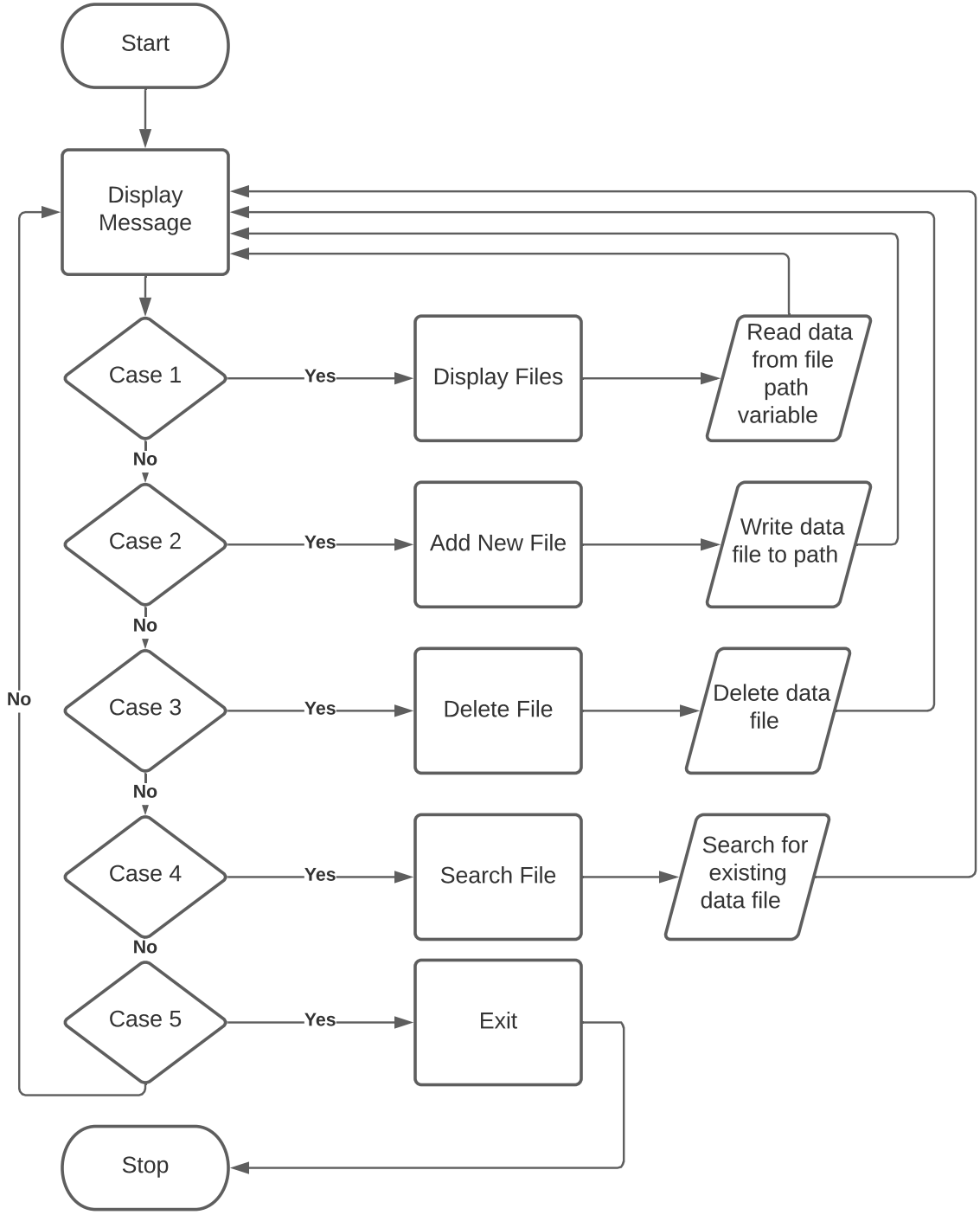


Figure 9 When inputting 4 an existing file name can be searched, and file will be found.

Implement OOPS using JAVA with Data Structures and Beyond