Arithmetic calculator

```
package com.practiceproject.solutions;
import java.util.Scanner;
public class ArithmeticCalculator {

	public static void main(String[] args) {


				Scanner sc=new Scanner(System.in);
				System.out.println("eneter the two numbers");
				int num1=sc.nextInt();
				int num2= sc.nextInt();
				System.out.println("Enter the operator ");
				char op=sc.next().charAt(0);
				double Ans=0;

				switch(op){
				case '+': Ans=num1+num2;
				break;
				case '-':Ans=num1-num2;
				break;
				case '*': Ans=num1*num2;
				break;
				case '/':Ans=num1/num2;
				break;
				}
				System.out.println("the answer is " +Ans);
		}


	}
```

Validation of an email

```
package com.practiceproject.solution;

import java.util.regex.*;
import java.util.*;
public class ValidationOfAnEmail {


	public static void main(String args[]){
	    ArrayList<String> emails = new ArrayList<String>();
	    emails.add("john123@domain.co.in");
	    emails.add("Amrita@domain.com");
	    emails.add("vinay.name@domain.com");
	    emails.add("arcot#@domain.co.in");
	    emails.add("kumar@domain.com");
	    emails.add("sai@domaincom");

	    //Add invalid email in list
```

```
                emails.add("@yahoo.com");
                emails.add("vinay#domain.com");

                //Regular Expression
                String regex = "^(.+)@(.+)$";

                //Compile regular expression to get the pattern
                Pattern pattern = Pattern.compile(regex);

                //Iterate Email array list
                for(String email : emails){
                    //Create instance of matcher
                    Matcher matcher = pattern.matcher(email);
                    System.out.println(email +" : "+ matcher.matches()+"\n");
                }
            }
}
```

File Handling:-

```
package com.practiceproject.solution;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
import java.util.StringTokenizer;

public class FileHandling {


             public static void main(String[] args) {
                    Scanner strInput = new Scanner(System.in);
                     String choice,cont = "y";

                     while( cont.equalsIgnoreCase("y") ) {
                             System.out.println("\t\t student Information
System\n\n");

                                 System.out.println("1 ===> Add New student Record ");
                                 System.out.println("2 ===> View All student Record ");
                                 System.out.println("3 ===> Delete student Record ");
                                 System.out.println("4 ===> Search Specific Record ");
                                 System.out.println("5 ===> Update Specific Record ");

                                 System.out.print("\n\n");
                                 System.out.println("Enter your choice: ");
                                 choice = strInput.nextLine();
```

```
                        if( choice.equals("1") ) {
                                try {
                                        AddRecord();
                                    } catch (IOException e) {

                                        e.printStackTrace();
                                    }
                        } else if( choice.equals("2") ) {
                                try {
                                        ViewAllRecord();
                                    } catch (IOException e) {

                                        e.printStackTrace();
                                    }
                        } else if( choice.equals("3") ) {
                                try {
                                        DeleteRecordByID();
                                    } catch (IOException e) {

                                        e.printStackTrace();
                                    }
                        }    else if( choice.equals("4") ) {
                                try {
                                        SearchRecordbyID();
                                    } catch (IOException e) {

                                        e.printStackTrace();
                                    }
                        }    else if( choice.equals("5") ) {
                                try {
                                        updateRecordbyID();
                                    } catch (IOException e) {

                                        e.printStackTrace();
                                    }
                        }

                        System.out.println("Do you want to continue? Y/N");
                        cont = strInput.nextLine();

                }

        }

          public static void AddRecord() throws IOException {

                BufferedWriter bw = new BufferedWriter( new
FileWriter("records.txt",true) );
                Scanner strInput = new Scanner(System.in);

                String ID, name, age, addr;

                System.out.print("Enter the student ID: ");
                ID = strInput.nextLine();
                System.out.print("Enter the student Name: ");
```

```
                    name = strInput.nextLine();
                    System.out.print("Enter the student Age: ");
                    age = strInput.nextLine();
                    System.out.print("Enter the student Address: ");
                    addr = strInput.nextLine();

                    bw.write(ID+","+name+","+age+","+addr);
                    bw.flush();
                    bw.newLine();
                    bw.close();

        }



            public static void ViewAllRecord() throws IOException {
                    BufferedReader br = new BufferedReader( new
FileReader("records.txt") );

                    String record;

                    System.out.println(" ------------------------------------------
---------------- ");
                    System.out.println("|     ID          Name
      Age             Address              |");
                    System.out.println(" ------------------------------------------
---------------- ");

                    while( ( record = br.readLine() ) != null ) {

                            StringTokenizer st = new StringTokenizer(record,",");

                            System.out.println("|     "+st.nextToken()+"
      "+st.nextToken()+"        "+st.nextToken()+"               "+st.nextToken()+"
|");

                    }

                    System.out.println("|
                |");
                    System.out.println(" ------------------------------------------
---------------- ");
                    br.close();

                }



            public static void DeleteRecordByID() throws IOException {
                    Scanner strInput =  new Scanner(System.in);
                    String ID, record;
```

```
                        File tempDB = new File("records_temp.txt");
                        File db = new File("records.txt");


                        BufferedReader br = new BufferedReader( new FileReader( db
) );
                        BufferedWriter bw = new BufferedWriter( new FileWriter(
tempDB ) );


                        System.out.println("\t\t Delete Employee Record\n");

                        System.out.println("Enter the Employee ID: ");
                        ID =  strInput.nextLine();


                        while( ( record = br.readLine() ) != null ) {


                              if( record.contains(ID) )
                                    continue;

                              bw.write(record);
                              bw.flush();
                              bw.newLine();

                        }

                        br.close();
                        bw.close();

                        db.delete();

                        tempDB.renameTo(db);

                  }


            public static void SearchRecordbyID() throws IOException {
                        String ID,record;
                        Scanner strInput = new Scanner(System.in);

                        BufferedReader br = new BufferedReader( new
FileReader("records.txt") );

                        System.out.println("\t\t Search student Record\n");


                        System.out.println("Enter the student ID: ");
                        ID = strInput.nextLine();

                        System.out.println(" ----------------------------------
---------------------- ");
                        System.out.println("|     ID          Name
      Age               Address              |");
```

```
                        System.out.println(" ------------------------------------
---------------------- ");

                        while( ( record = br.readLine() ) != null ) {

                                StringTokenizer st = new
StringTokenizer(record,",");
                                if( record.contains(ID) ) {
                                        System.out.println("|     "+st.nextToken()+"
      "+st.nextToken()+"        "+st.nextToken()+"               "+st.nextToken()+"
|");
                                }



                        }

                        System.out.println("|
                |");
                        System.out.println(" ------------------------------------
---------------------- ");

                        br.close();



                }


            public static void updateRecordbyID() throws IOException {
                        String newName, newAge, newAddr, record, ID,record2;

                        File db = new File("records.txt");
                        File tempDB = new File("records_temp.txt");

                        BufferedReader br = new BufferedReader( new FileReader(db)
);
                        BufferedWriter bw = new BufferedWriter( new
FileWriter(tempDB) );

                        Scanner strInput = new Scanner(System.in);

                        System.out.println("\t\t Update student Record\n\n");
                        /**/
                                System.out.println("Enter the student ID: ");
                                ID = strInput.nextLine();
                                System.out.println(" ------------------------------
---------------------------- ");
                                System.out.println("|     ID          Name
            Age               Address               |");
                                System.out.println(" ------------------------------
---------------------------- ");

                                while( ( record = br.readLine() ) != null ) {
```

```
            StringTokenizer st = new StringTokenizer(record,",");
                        if( record.contains(ID) ) {
                       System.out.println("| "+st.nextToken()+"  "+st.nextToken()+"
      "+st.nextToken()+"               "+st.nextToken()+"      |");
                                        }

                                  }
                  System.out.println("|
|");
                                  System.out.println(" ------------------------------
---------------------------- ");

                           br.close();
                           /**/
                           System.out.println("Enter the new Name: ");
                           newName = strInput.nextLine();
                           System.out.println("Enter the new Age: ");
                           newAge = strInput.nextLine();
                           System.out.println("Enter the new Address: ");
                           newAddr = strInput.nextLine();

                  BufferedReader br2 = new BufferedReader( new FileReader(db) );

                           while( (record2 = br2.readLine() ) != null ) {

                                  if(record2.contains(ID)) {

      bw.write(ID+","+newName+","+newAge+","+newAddr);
                                  } else {

                                        bw.write(record2);
                                  }
                                  bw.flush();
                                  bw.newLine();
                           }

                           bw.close();
                           br2.close();
                           db.delete();
                           boolean success = tempDB.renameTo(db);
                           System.out.println(success);

                  }


      }
```

Bugs fix:-

```java
package com.practiceproject.solutions;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.Scanner;
public class BugsFix {



    public static void main(String[] args) {
      System.out.println("Hello Howdy!");
      System.out.println("\n-------------\n");
      System.out.println("\tWelcome to TheDesk \n");
      System.out.println("-----------");
      optionsSelection();

   }
   private static void optionsSelection() {
       String[] arr = {"1. I wish to review my application",
               "2. I want to add my data",
               "3. I want to delete my data",
               "4. I want to sort the data",
               "5. Close the application"
       };
       int[] arr1 = {1,2,3,4,5,};
       int  slen = arr1.length;
       for(int i=0; i<slen;i++){
           System.out.println(arr[i]);
           // display the all the Strings mentioned in the String array
       }
       ArrayList<Integer> arrlist = new ArrayList<Integer>();
       ArrayList<Integer> expenses = new ArrayList<Integer>();
       expenses.add(101);
       expenses.add(2021);
       expenses.add(20220);
       expenses.add(40000);
       expenses.add(10000);
       expenses.addAll(arrlist);
       System.out.println("\nEnter your choice:\t");
       Scanner sc = new Scanner(System.in);
       int  options =  sc.nextInt();
       for(int j=1;j<=slen;j++){
           if(options==j){
               switch (options){
                   case 1:
                  System.out.println("Your saved data are listed below: \n");
                        System.out.println(expenses+"\n");
                        optionsSelection();
                        break;
                    case 2:
                  System.out.println("Enter the value to add your data: \n");
                        int value = sc.nextInt();
                        expenses.add(value);
```

```
                                System.out.println("Your value is updated\n");
                                expenses.addAll(arrlist);
                                System.out.println(expenses+"\n");
                                optionsSelection();

                                break;
                            case 3:
                                System.out.println("You are about the delete all your
data! \nConfirm again by selecting the same option...\n");
                                int con_choice = sc.nextInt();
                                if(con_choice==options){
                                       expenses.clear();
                                    System.out.println(expenses+"\n");
                                    System.out.println("All your data are erased!\n");
                                } else {
                                    System.out.println("Oops... try again!");
                                }
                                optionsSelection();
                                break;
                            case 4:
                                sortdata(expenses);
                                optionsSelection();
                                break;
                            case 5:
                                searchdata(expenses);
                                optionsSelection();
                                break;
                            case 6:
                                closeApp();
                                break;
                            default:
                               System.out.println("You have made an invalid choice!");
                                break;
                        }
                    }
                }

            }
            private static void closeApp() {
                System.out.println("Closing your application... \nThank you!");
                    }
            private static void searchdata(ArrayList<Integer> arrayList) {
                int leng = arrayList.size();
                System.out.println("Enter the data you need to search:\t");
                //
                Scanner sc = new Scanner(System.in);
                int input = sc.nextInt();
                //Linear Search
                for(int i=0;i<leng;i++) {
                     if(arrayList.get(i)==input) {
             System.out.println("Found the data " + input + " at " + i + " position");
                    }
                }
            }
            private static void sortdata(ArrayList<Integer> arrayList) {
```

```
            int arrlength =  arrayList.size();
           //Complete the method. The data should be sorted in ascending order.

            Collections.sort(arrayList);
            System.out.println("Sorted data: ");
            for(Integer i: arrayList) {
                 System.out.print(i + " ");
            }

            System.out.println("\n");

        }
    }
```

Longest Increasing subsequence:-

```
package com.practiceproject.solution;

public class LongestIncreasingSubsequence {

    static int max_ref;

    static int _lis(int arr[], int n)
    {

        if (n == 1)
            return 1;

        int res, max_ending_here = 1;


        for (int i = 1; i < n; i++) {
            res = _lis(arr, i);
            if (arr[i - 1] < arr[n - 1]
                && res + 1 > max_ending_here)
                max_ending_here = res + 1;
        }


        if (max_ref < max_ending_here)
            max_ref = max_ending_here;


        return max_ending_here;
    }


    static int lis(int arr[], int n)
    {
```

```
        max_ref = 1;


        _lis(arr, n);


        return max_ref;
    }


    public static void main(String args[])
    {
        int arr[] = { 5,10,3,15,38,45,9,65,74,33,80 };
        int n = arr.length;
        System.out.println("Length of lis is " + lis(arr, n)
                                    + "\n");

    }
}
```