NAME: Pavithra R

DATE: 23/07/2025

DAY 13 ASSESSMENT

1)1. A library needs to develop an online application for two types of users/roles, Adults and children. Both of these users should be able to register an account. Any user who is less than 12 years of age will be registered as a child and they can borrow a "Kids" category book for 10 days, whereas an adult can borrow "Fiction" category books which need to be returned within 7 days.

1.Create an interface LibraryUser with the following methods declared,

Method Name

registerAccount

requestBook

2.Create 2 classes "KidUser" and "AdultUser" which implements the LibraryUser interface.

3.Both the classes should have two instance variables as specified below.

age	int
bookType	String

4.The methods in the KidUser class should perform the following logic.

1. registerAccount : if age < 12, a message displaying “You have successfully registered under a Kids Account” should be displayed in the console.

If(age>12), a message displaying, "Sorry, Age must be less than 12 to register as a kid" should be displayed in the console.

2. requestBook : if bookType is "Kids", a message displaying "Book Issued successfully, please return the book within 10 days" should be displayed in the console.

else, a message displaying, "You are allowed to take only kids books" should be displayed in the console.

5.The methods in the AdultUser class should perform the following logic.

1. registerAccount : if age > 12, a message displaying "You have successfully registered under an Adult Account" should be displayed in the console.

If age<12, a message displaying, "Sorry, Age must be greater than 12 to register as an adult" should be displayed in the console.

2. requestBook : if bookType is "Fiction", a message displaying "Book Issued successfully, please return the book within 7 days" should be displayed in the console.

else, a message displaying, "

You are allowed to take only adult Fiction books" should be displayed in the console.

6.Create a class LibraryInterfaceDemo with a main method which performs the below functions,

In the main method, test all the methods.

CODE:

```
import java.util.*;


interface LibraryUser {
    void registerAccount();
    void requestBook();
}


class KidUser implements LibraryUser {
    int age;
    String bookType;


    // Constructor
```

```
    public KidUser(int age, String bookType) {
        this.age = age;
        this.bookType = bookType;
    }

    public void registerAccount() {
        if (age < 12) {
            System.out.println("You have successfully registered under a Kids Account");
        } else {
            System.out.println("Sorry, Age must be less than 12 to register as a kid");
        }
    }

    public void requestBook() {
        if (bookType.equalsIgnoreCase("Kids")) {
            System.out.println("Book Issued successfully, please return the book within 10 days");
        } else {
            System.out.println("You are allowed to take only kids books");
        }
    }
}

class AdultUser implements LibraryUser {
    int age;
    String bookType;
```

```
    public AdultUser(int age, String bookType) {
        this.age = age;
        this.bookType = bookType;
    }

    public void registerAccount() {
        if (age > 12) {
            System.out.println("You have successfully registered under an Adult Account");
        } else {
            System.out.println("Sorry, Age must be greater than 12 to register as an adult");
        }
    }

    public void requestBook() {
        if (bookType.equalsIgnoreCase("Fiction")) {
            System.out.println("Book Issued successfully, please return the book within 7 days");
        } else {
            System.out.println("You are allowed to take only adult Fiction books");
        }
    }
}

public class Main {
    public static void main(String[] args) {

        System.out.println("----- Kid User 1 -----");
```

```
        KidUser kid1 = new KidUser(10, "Kids");
        kid1.registerAccount();
        kid1.requestBook();

        System.out.println("----- Kid User 2 -----");
        KidUser kid2 = new KidUser(14, "Fiction");
        kid2.registerAccount();
        kid2.requestBook();

        System.out.println("----- Adult User 1 -----");
        AdultUser adult1 = new AdultUser(25, "Fiction");
        adult1.registerAccount();
        adult1.requestBook();

        System.out.println("----- Adult User 2 -----");
        AdultUser adult2 = new AdultUser(10, "Kids");
        adult2.registerAccount();
        adult2.requestBook();
    }
}
```

OUTPUT:

```
----- Kid User 1 -----
You have successfully registered under a Kids Account
Book Issued successfully, please return the book within 10 days
----- Kid User 2 -----
Sorry, Age must be less than 12 to register as a kid
You are allowed to take only kids books
----- Adult User 1 -----
You have successfully registered under an Adult Account
Book Issued successfully, please return the book within 7 days
----- Adult User 2 -----
Sorry, Age must be greater than 12 to register as an adult
You are allowed to take only adult Fiction books
```

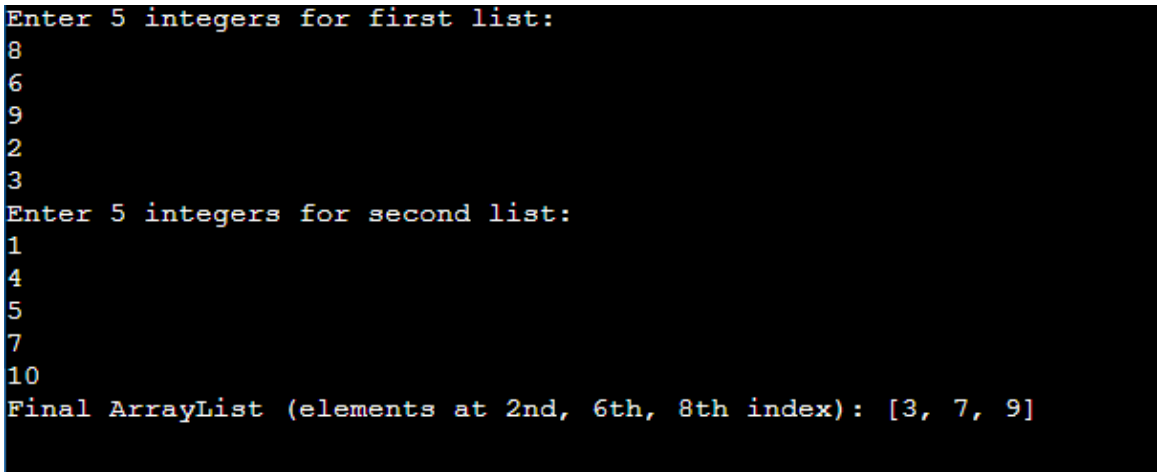
2)Write a program to read two integer array lists of size 5 each as input and to merge the two arrayLists, sort the merged arraylist in ascending order and fetch the elements at 2nd, 6th and 8th index into a new arrayList and return the final ArrayList.

CODE:

```
import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;


public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


        ArrayList<Integer> list1 = new ArrayList<>();

        ArrayList<Integer> list2 = new ArrayList<>();


        System.out.println("Enter 5 integers for first list:");

        for (int i = 0; i < 5; i++) {

            list1.add(scanner.nextInt());

        }
```

```
        System.out.println("Enter 5 integers for second list:");
        for (int i = 0; i < 5; i++) {
            list2.add(scanner.nextInt());
        }
        ArrayList<Integer> mergedList = new ArrayList<>(list1);
        mergedList.addAll(list2);
        Collections.sort(mergedList);
        ArrayList<Integer> resultList = new ArrayList<>();
        if (mergedList.size() > 8) {
            resultList.add(mergedList.get(2));
            resultList.add(mergedList.get(6));
            resultList.add(mergedList.get(8));
        } else {
            System.out.println("Not enough elements to fetch from merged list.");
        }

        System.out.println("Final ArrayList (elements at 2nd, 6th, 8th index): " + resultList);

        scanner.close();
    }
}
```

OUTPUT:



3)Read student details as input.The details would include name, mark in the given order. The datatype for name is string, mark is float.Create a hashmap that contains name as key and mark as value.Get student name as input and display the student grade.

1. If Mark is less than 60, then grade is FAIL.

2. If Mark is greater than or equal to 60, then grade is PASS

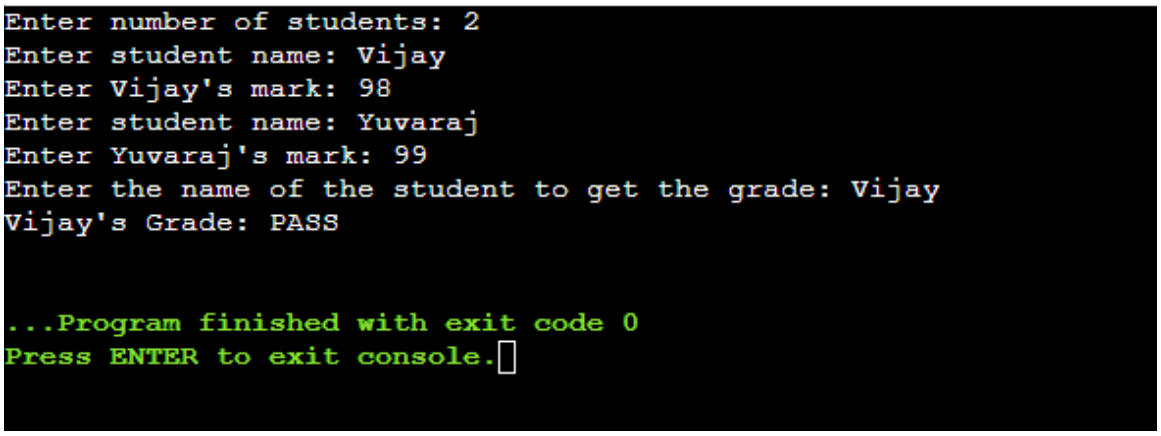
CODE:

```
import java.util.HashMap;

import java.util.Scanner;


public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        HashMap<String, Float> studentMap = new HashMap<>();


        System.out.print("Enter number of students: ");

        int n = scanner.nextInt();

        scanner.nextLine();


        for (int i = 0; i < n; i++) {
```

```
        System.out.print("Enter student name: ");
        String name = scanner.nextLine();

        System.out.print("Enter " + name + "'s mark: ");
        float mark = scanner.nextFloat();
        scanner.nextLine();

        studentMap.put(name, mark);
    }


System.out.print("Enter the name of the student to get the grade: ");
String searchName = scanner.nextLine();

if (studentMap.containsKey(searchName)) {
        float mark = studentMap.get(searchName);
        if (mark >= 60) {
            System.out.println(searchName + "'s Grade: PASS");
        } else {
            System.out.println(searchName + "'s Grade: FAIL");
        }
} else {
        System.out.println("Student not found.");
}

scanner.close();
```

```
    }
}
```

OUTPUT:



4)Write a program to get integers as input and store in the arraylist. Traverse the input list, if the number is even store in a arraylist called evenNumbersList and oddnumbers in oddNumberList. Print the input list and the lists containing even numbers and odd numbers.

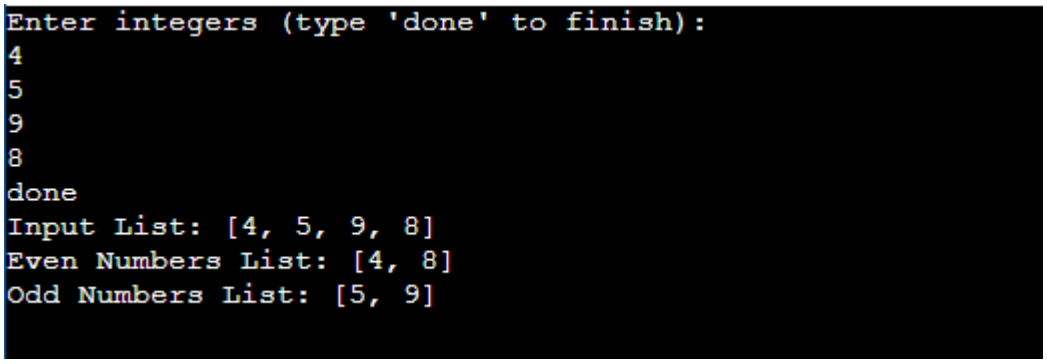
CODE:

```
import java.util.ArrayList;

import java.util.Scanner;


public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


        ArrayList<Integer> inputList = new ArrayList<>();

        ArrayList<Integer> evenNumbersList = new ArrayList<>();

        ArrayList<Integer> oddNumbersList = new ArrayList<>();


        System.out.println("Enter integers (type 'done' to finish):");
```

```
        while (scanner.hasNext()) {
            if (scanner.hasNextInt()) {
                int number = scanner.nextInt();
                inputList.add(number);
            } else {
                String input = scanner.next();
                if (input.equalsIgnoreCase("done")) {
                    break;
                } else {
                    System.out.println("Invalid input, enter an integer or 'done' to finish:");
                }
            }
        }
        for (int num : inputList) {
            if (num % 2 == 0) {
                evenNumbersList.add(num);
            } else {
                oddNumbersList.add(num);
            }
        }
        System.out.println("Input List: " + inputList);
        System.out.println("Even Numbers List: " + evenNumbersList);
        System.out.println("Odd Numbers List: " + oddNumbersList);

        scanner.close();
    }
```

}

OUTPUT:


Enter integers (type 'done' to finish):
4
5
9
8
done
Input List: [4, 5, 9, 8]
Even Numbers List: [4, 8]
Odd Numbers List: [5, 9]