

# **CSE1007 – JAVA PROGRAMMING**

## **DIGITAL ASSIGNMENT – 1**

**NAME : RUSHABH KELA**

**REG. NO. : 19BDS0055**

**SLOT : C2 + TC2**

**FACULTY : Prof. JAISANKAR N**

**SCHOOL : SCOPE, VIT**

**DATE : JUNE 3, 2021**

## 1. ARRAY –

**COVID Vaccination drive has started in Maharashtra. Take input from user for the name of districts, the number of vaccination centres in that district people registered for vaccination. and store in a 2D array. Create a jagged array in which each row is a district and columns is number of vaccination centres in which maximum 40 people can be vaccinated per day. Display the number of people allotted to each vaccination centre and check if all people registered on that day were vaccinated or not.**

**Example :**

District	Number of vaccination centres	Number of people registered
Nagpur	5	100
Mumbai	10	500
Pune	3	120

**Output :**

**Nagpur : 40 40 20 0 0**

**Mumbai : 40 40 40 40 40 40 40 40 40 40**

**Pune : 40 40 40**

**Only 400 people were able to vaccinate in Mumbai and 100 people were left. Nagpur had 3 centres free on that day.**

**CODE :**

```
import java.util.*;
import java.io.*;
import java.lang.*;

public class DistrictVaccine {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n,i,j,k,p;

        System.out.println("Maharashtra COVID-19 Vaccination Drive");
        System.out.print("Enter the number of districts: ");
        n = sc.nextInt();

        String[] district = new String[n];
```

```

int[][] centres = new int[n][];
boolean[] free = new boolean[n];
int[] left = new int[n];
System.out.println("\nEnter the details of each district: ");
for(i=0;i<n;i++)
{
    free[i] = false;
    System.out.println("\nDistrict " + (i+1));
    System.out.print("Enter Name: ");
    district[i] = sc.next();
    System.out.print("Enter number of vaccination centres: ");
    k = sc.nextInt();
    System.out.print("Enter number of people registered: ");
    p = sc.nextInt();

    centres[i] = new int[k];
    for(j=0;j<k;j++)
    {
        if(p > 40) {
            centres[i][j] = 40;
            p -= 40;
        }
        else if (p > 0) {
            centres[i][j] = p;
            p -= p;
        }
        else {
            centres[i][j] = 0;
            free[i] = true;
        }
    }
    left[i] = p;
}

// Traversing the jagged array using foreach loop
System.out.println();
k = 0;
for (int[] cv : centres) {
    System.out.print(district[k++] + ": ");
    for (int num : cv) {
        System.out.print(num + " ");
    }
    System.out.println();
}

```

```
}

// Printing the districts where all people were not able to vaccinate
k=0;
for (int[] cv : centres) {
    if(left[k] != 0) {
        System.out.println("Only " + 40*cv.length + " people were able to vaccinate
in " + district[k] + " and " + left[k] + " people were left.");
    }
    k++;
}

// Districts with free centres
k=0;
for (int[] cv : centres) {
    int cnt = 0;
    if(free[k]) {
        for (int num : cv) {
            if(num != 0) {
                cnt++;
            }
        }
        System.out.println(district[k] + " had " + cnt + " centres free on that day.");
    }
    k++;
}
}
```

## OUTPUT:

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac DistrictVaccine.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java DistrictVaccine
Maharashtra COVID-19 Vaccination Drive
Enter the number of districts: 3

Enter the details of each district:

District 1
Enter Name: Nagpur
Enter number of vaccination centres: 5
Enter number of people registered: 100

District 2
Enter Name: Mumbai
Enter number of vaccination centres: 10
Enter number of people registered: 500

District 3
Enter Name: Pune
Enter number of vaccination centres: 3
Enter number of people registered: 120

Nagpur: 40 40 20 0 0
Mumbai: 40 40 40 40 40 40 40 40 40
Pune: 40 40 40
Only 400 people were able to vaccinate in Mumbai and 100 people were left.
Nagpur had 3 centres free on that day.
```

## **2. STRING –**

**Rushabh was solving questions on palindromes. He has n distinct strings of equal length m. He wants to make the longest palindrome possible concatenating as many strings as possible. Write a Java program given the values of n, m, and the strings and print the longest palindromic string possible.**

### **CODE :**

```
import java.util.*;
import java.io.*;
import java.lang.*;

public class LongestPalindrome {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter n: ");
        int n = sc.nextInt();
        System.out.print("Enter m: ");
        int m = sc.nextInt();
        String arr1[] = new String[n + 10];
        String arr2[] = new String[n + 10];
        String longestPalString = "";

        System.out.println("Enter the strings: ");
        for (int i = 1; i <= n; i++)
        {
            String a = sc.next();
            String b = new StringBuffer(a).reverse().toString();
            arr1[i] = a;
            arr2[i] = b;
            if (arr1[i].equals(arr2[i]))
            {
                longestPalString = arr1[i];
            }
        }
        for (int i = 1; i <= n; i++)
        {
            for (int j = i + 1; j <= n; j++)
            {
                if (arr2[i].equals(arr1[j]))
                    longestPalString = arr1[i] + longestPalString + arr2[i];
            }
        }
    }
}
```

```
        }
    }
    System.out.println("Longest palindrome length is : " + longestPalString.length());
    System.out.println("Longest Palindrome: " + longestPalString);
}
}
```

## OUTPUT :

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac LongestPalindrome.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java LongestPalindrome
Enter n: 3
Enter m: 3
Enter the strings:
tab
one
bat
Longest palindrome length is : 6
Longest Palindrome: tabbat
```

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java LongestPalindrome
Enter n: 9
Enter m: 4
Enter the strings:
abab
baba
abcd
bcde
cdef
defg
wxyz
zyxw
ijji
Longest palindrome length is : 20
Longest Palindrome: wxyzababijjibabazyw
```

### **3. INHERITANCE –**

**Create a class Student to store the name, register number, email and proctor name of the student. Create another class Marks that inherit the student class and stores all the CAT1, CAT2, Quiz 1, Quiz 2, DA1, and FAT marks of the student. Create another class Result that inherits the marks class and has methods to calculate the average and SD of the class. Initialize all classes with appropriate constructor. Determine the grades of the student according to grading system based on average and SD.**

### **CODE :**

```
import java.util.*;
import java.io.*;
import java.lang.*;

class Student {
    String name, regNo;
    long mobile;
    String email, proctor_name;

    Student(String name, String regNo, long mobile, String email, String proctor_name)
    {
        this.name = name;
        this.regNo = regNo;
        this.mobile = mobile;
        this.email = email;
        this.proctor_name = proctor_name;
    }

}

class Marks extends Student {
    double cat1;
    double cat2;
    double da1;
    double quiz1;
    double quiz2;
    double fat;
    Marks(String name, String regNo, long mobile, String email, String proctor_name,
double cat1, double cat2, double quiz1, double quiz2, double da1, double fat) {
        super(name, regNo, mobile, email, proctor_name);
```

```

        this.cat1 = cat1;
        this.cat2 = cat2;
        this.quiz1 = quiz1;
        this.quiz2 = quiz2;
        this.da1 = da1;
        this.fat = fat;
    }
}

class Result extends Marks {
    double total;
    Result(String name, String regNo, long mobile, String email, String proctor_name,
double cat1, double cat2, double quiz1, double quiz2, double da1, double fat) {
        super(name, regNo, mobile, email, proctor_name, cat1, cat2, quiz1, quiz2, da1,
fat);
        this.total = (cat1/50)*15 + (cat2/50)*15 + quiz1 + quiz2 + da1 + (fat/100)*40;
    }
    static double calcAvg(Result[] arr) {
        double sum = 0, avg = 0;
        for(Result r: arr) {
            sum += r.total;
        }
        avg = sum / arr.length;
        return avg;
    }
    static double calcSD(Result[] arr) {
        double sum = 0, sd;
        double u = calcAvg(arr);
        for(Result r: arr) {
            sum += Math.pow((r.total - u), 2);
        }
        sd = Math.sqrt(sum / arr.length);
        return sd;
    }
}

void display(String grade) {
    System.out.println("\nResult of Student " + this.name + " (" + this.regNo + ")");
    System.out.println("Mobile Number: " + this.mobile);
    System.out.println("Email: " + this.email);
    System.out.println("Proctor Name: " + this.proctor_name);
    System.out.println("TOTAL MARKS OBTAINED: " + this.total);
    System.out.println("GRADE: " + grade);
}

```

```
}

public class Grades {
    public static void main(String[] args) {
        System.out.println("-----");
        System.out.println("|\tVIT GRADING SYSTEM\t|");
        System.out.println("-----");

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students in the class: ");
        int n = sc.nextInt();
        int i;
        String name, regNo, email, proctor_name;
        long mobile;
        double cat1, cat2, quiz1, quiz2, da1, fat;

        Result[] slot = new Result[n];
        for(i=0;i<n;i++)
        {
            System.out.println("\nEnter Student Details: ");
            System.out.print("Enter Name: ");
            name = sc.next() + sc.nextLine();
            System.out.print("Enter Register Number: ");
            regNo = sc.next();
            System.out.print("Enter Mobile: ");
            mobile = sc.nextLong();
            System.out.print("Enter EMail: ");
            email = sc.next();
            sc.nextLine();
            System.out.print("Enter Proctor Name: ");
            proctor_name = sc.nextLine();
            System.out.println("Mark Entry: ");
            System.out.print("CAT1 (out of 50): ");
            cat1 = sc.nextDouble();
            System.out.print("CAT2 (out of 50): ");
            cat2 = sc.nextDouble();
            System.out.print("Digital Assignment: ");
            da1 = sc.nextDouble();
            System.out.print("Quiz 1: ");
            quiz1 = sc.nextDouble();
            System.out.print("Quiz 2: ");
            quiz2 = sc.nextDouble();
            System.out.print("Final Assessment Test (out of 100): ");
            fat = sc.nextDouble();
        }
    }
}
```

```

slot[i] = new Result(name, regNo, mobile, email, proctor_name, cat1, cat2,
quiz1, quiz2, da1, fat);
}

double avg = Result.calcAvg(slot);
avg = Math.round(avg*100)/100;
double sd = Result.calcSD(slot);
sd = Math.round(sd*100)/100;
System.out.println("\nCLASS RESULT");
System.out.println("Total number of students: " + slot.length);
System.out.println("Class Average: " + avg);
System.out.println("SD: " + sd);

for(Result r: slot) {
    if(r.total >= avg + 1.5*sd) {
        r.display("S");
    }
    else if(r.total >= avg + 0.5*sd) {
        r.display("A");
    }
    else if(r.total >= avg - 0.5*sd) {
        r.display("B");
    }
    else if(r.total >= avg - 1*sd) {
        r.display("C");
    }
    else if(r.total >= avg - 1.5*sd) {
        r.display("D");
    }
    else if(r.total >= avg - 2*sd) {
        r.display("E");
    }
    else {
        r.display("F");
    }
}
}

```

## OUTPUT:

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Grades.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Grades
-----
|          VIT GRADING SYSTEM          |
-----
Enter the number of students in the class: 3

Enter Student Details:
Enter Name: Rushabh Kela
Enter Register Number: 19BDS0055
Enter Mobile: 9834473257
Enter EMail: rushabhbhagwandas.kela2019@vitstudent.ac.in
Enter Proctor Name: Prof. Raju R L N
Mark Entry:
CAT1 (out of 50): 45
CAT2 (out of 50): 45
Digital Assignment: 10
Quiz 1: 9
Quiz 2: 10
Final Assessment Test (out of 100): 90

Enter Student Details:
Enter Name: Abhinav Chawla
Enter Register Number: 19BCE0343
Enter Mobile: 8200476905
Enter EMail: abhinav.chawla2019@vitstudent.ac.in
Enter Proctor Name: Prof. Deepika S
Mark Entry:
CAT1 (out of 50): 43.5
CAT2 (out of 50): 44
Digital Assignment: 10
Quiz 1: 9
Quiz 2: 9
Final Assessment Test (out of 100): 88

Enter Student Details:
Enter Name: Ashutosh
Enter Register Number: 19BCE0134
Enter Mobile: 7722092228
Enter EMail: ashutosh.2019@vitstudent.ac.in
Enter Proctor Name: Prof. Seetha R
Mark Entry:
CAT1 (out of 50): 40
CAT2 (out of 50): 42
```

```
Digital Assignment: 10
Quiz 1: 9
Quiz 2: 8
Final Assessment Test (out of 100): 80
```

#### CLASS RESULT

```
Total number of students: 3
Class Average: 88.35
SD: 3.52
```

Result of Student Rushabh Kela (19BDS0055):

```
Mobile Number: 9834473257
Email: rushabhbhagwandas.kela2019@vitstudent.ac.in
Proctor Name: Prof. Raju R L N
TOTAL MARKS OBTAINED: 92.0
GRADE: A
```

Result of Student Abhinav Chawla (19BCE0343):

```
Mobile Number: 8200476905
Email: abhinav.chawla2019@vitstudent.ac.in
Proctor Name: Prof. Deepika S
TOTAL MARKS OBTAINED: 89.45
GRADE: B
```

Result of Student Ashutosh (19BCE0134):

```
Mobile Number: 7722092228
Email: ashutosh.2019@vitstudent.ac.in
Proctor Name: Prof. Seetha R
TOTAL MARKS OBTAINED: 83.6
GRADE: D
```

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$
```

#### **4. INTERFACES –**

**Create interface security with the data member length to be fixed initially. Declare new function such as encrypt(string) and decrypt(string). Create another interface call stringfun which is inherited from the interface security. Create a class pwdCheck that reads the input and converts it to the encrypted form and vice-versa.**

#### **CODE :**

```
import java.util.*;
import java.io.*;
import java.lang.*;

interface security {
    int length = 8;
    void encrypt(String s);
    void decrypt(String s);
}

interface stringfun extends security {
    void read();
    void display(boolean encryptOperation, String s);
}

class PwdCheck implements stringfun {
    String password;
    Scanner sc = new Scanner(System.in);
    public void encrypt(String s) {
        StringBuffer sb = new StringBuffer(s);
        int i = 0;
        for(i=0;i<s.length();i++) {
            int temp = s.charAt(i);
            temp++;
            char encryptChar = (char)temp;
            sb.setCharAt(i, encryptChar);
        }
        this.password = sb.toString();
    }

    public void decrypt(String s) {
        StringBuffer sb = new StringBuffer(s);
        int i = 0;
        for(i=0;i<s.length();i++) {
            int temp = s.charAt(i);
```

```

        temp--;
        char decryptChar = (char)temp;
        sb.setCharAt(i, decryptChar);
    }
    this.password = sb.toString();
}

public void read() {
    System.out.print("Enter Password: ");
    password = sc.next();
    System.out.println("Password Saved.");
}

public void display(boolean b, String s) {
    if(b) {
        System.out.println("Encrypted Password: " + s);
    }
    else {
        System.out.println("Decrypted Password: " + s);
    }
}
}

public class Encryption {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("--- Data Security ---");
        int ch;
        do {
            System.out.println("\nEnter the operation you want to perform: ");
            System.out.println("1. Encrypt Password");
            System.out.println("2. Decrypt Password");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            ch = sc.nextInt();
            switch(ch)
            {
                case 1:
                    PwdCheck enc = new PwdCheck();
                    enc.read();
                    enc.encrypt(enc.password);
                    enc.display(true, enc.password);
                    break;
            }
        }
    }
}

```

```

        case 2:
            PwdCheck dec = new PwdCheck();
            dec.read();
            dec.decrypt(dec.password);
            dec.display(false, dec.password);
            break;

        case 3:
            System.out.println("Thank You.");
        }
    }while(ch!=3);
}
}

```

## **OUTPUT:**

```

[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Encryption.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Encryption
--- Data Security ---

Enter the operation you want to perform:
1. Encrypt Password
2. Decrypt Password
3. Exit
Enter your choice: 1
Enter Password: RushabhKela45
Password Saved.
Encrypted Password: SvtibciLfmb56

Enter the operation you want to perform:
1. Encrypt Password
2. Decrypt Password
3. Exit
Enter your choice: 2
Enter Password: SvtibciLfmb56
Password Saved.
Decrypted Password: RushabhKela45

Enter the operation you want to perform:
1. Encrypt Password
2. Decrypt Password
3. Exit
Enter your choice: 3
Thank You.
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$
```

## 5. PACKAGES –

Rushabh was learning Pythagorean Theorem of right angle triangle with side a,b,c satisfying the formula  $a^2 + b^2 = c^2$ . However during his exam, he forgot the actual formula and found the Pythagorean triplets using the incorrect formula,  $c = a^2 - b$ . Given an integer, n, find the Pythagorean triplets such that  $1 \leq a \leq b \leq c \leq n$  and they satisfy both the conditions. Within the package ‘correctformula’, define a class with method to calculate all the correct Pythagorean triplets. Define another class ‘rushabhfomula’ outside of this package to check the if the correct triplets also satisfy his condition. Print them.

**Example : (a,b,c) = (3,4,5).**

Here,  $3^2 + 4^2 = 5^2$  and also  $5 = 3^2 - 4$

- **correctformula**
  - **PythagoreanTriplets.java**

```
package correctformula;
```

```
public class PythagoreanTriplets {  
    public static int[][] correct(int n)  
    {  
        int a,b,c;  
        int[][] correct_triplets = new int[n*n][3];  
        int k=0;  
        for(a = 1; a <= n; a++)  
        {  
            for(b = a; b <= n; b++)  
            {  
                for(c = b; c <= n; c++)  
                {  
                    if(Math.pow(a,2) + Math.pow(b,2) == Math.pow(c, 2))  
                    {  
                        correct_triplets[k][0] = a;  
                        correct_triplets[k][1] = b;  
                        correct_triplets[k++][2] = c;  
                    }  
                }  
            }  
        }  
        return correct_triplets;  
    }  
}
```

- **rushabhformula**
  - **RushabhTriplets.java**

```
package rushabhformula;

public class RushabhTriplets {
    public static int[][] rushabh(int n)
    {
        int a,b,c;
        int[][] rushabh_triplets = new int[n*n][3];
        int k=0;
        for(a = 1; a <= n; a++)
        {
            for(b = a; b <= n; b++)
            {
                for(c = b; c <= n; c++)
                {
                    if(c == Math.pow(a,2) - b)
                    {
                        rushabh_triplets[k][0] = a;
                        rushabh_triplets[k][1] = b;
                        rushabh_triplets[k++][2] = c;
                    }
                }
            }
        }
        return rushabh_triplets;
    }
}
```

- **Mathematics.java**

```
import java.util.*;
import java.io.*;
import java.lang.*;

import correctformula.*;
import rushabhformula.*;

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

```

Scanner sc = new Scanner(System.in);
System.out.print("Enter n: ");
int n = sc.nextInt();

System.out.println("Pythagorean Triplets that satisfy both a^2 + b^2 = c^2 and c
= a^2 - b are");
int[][] correct = PythagoreanTriplets.correct(n);
int[][] rushabh = RushabhTriplets.rushabh(n);

ans:
for (int[] ctr : rushabh) {
    for (int[] rtr : correct) {
        if(Arrays.equals(ctr, new int[]{0,0,0})) {
            break ans;
        }
        if(Arrays.equals(ctr, rtr))
        {
            System.out.println("(" + ctr[0] + ", " + ctr[1] + ", " + ctr[2] + ")");
        }
    }
}
}

```

## OUTPUT :

```

|Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Mathematics.java
|Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Mathematics
Enter n: 30
Pythagorean Triplets that satisfy both a^2 + b^2 = c^2 and c = a^2 - b are
(3, 4, 5)
(5, 12, 13)
(7, 24, 25)

```

```

|Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ 
|Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Mathematics
Enter n: 100
Pythagorean Triplets that satisfy both a^2 + b^2 = c^2 and c = a^2 - b are
(3, 4, 5)
(5, 12, 13)
(7, 24, 25)
(9, 40, 41)
(11, 60, 61)
(13, 84, 85)

```

## 6. USER – DEFINED EXCEPTIONS :

Create a class by name Employee with members – Employee ID, Name and year of birth. The Employee ID is a string that contains the ID in the format year-designation-number. The year is represented with the last two digits. The designation is a single letter code - 'F' for faculty and 'S' for staff. The number is a 3 digit number.

(Example: 81-F-112 55-S-254)

Write a program to read the employee details and validate the employee code. If the employee code is incorrect throw a user-defined exception "InvalidEmployeeCode" else create the Employee object and display the details of the employee. Also check if the employee is over 65 years of age, he needs to retire and a bonus equal to his *age × number on his ID* will be given to him from the company.

Example : The employee 55-S-254 must retire since his current age is 66 and bonus of (66\*254) 13,970 will be given to him.

### CODE :

```
import java.util.*;
import java.io.*;
import java.lang.*;

class Employee {
    String empID;
    String name;
    int yearOfBirth;
    char designation;

    Employee(String name, String empID, int yearOfBirth, char designation) {
        this.empID = empID;
        this.name = name;
        this.yearOfBirth = yearOfBirth;
        this.designation = designation;
    }

    void display() {
        System.out.println("\nEmployee Details: ");
        System.out.println("Name: " + this.name);
        System.out.println("Employee ID: " + this.empID);
        System.out.println("Year of Birth: " + this.yearOfBirth);
        if(Character.compare(this.designation, 'F') == 0) {
            System.out.println("Designation: Faculty");
        }
    }
}
```

```

        }
    else {
        System.out.println("Designation: Staff");
    }
}
}

class InvalidEmployeeCode extends Exception {
    InvalidEmployeeCode(String s) {
        super(s);
    }
}

class InvalidAgeException extends Exception {
    InvalidAgeException(String s) {
        super(s);
    }
}

public class EmpDetails {
    static void validate(String empID, int yearOfBirth, char designation) throws
InvalidEmployeeCode
    {
        String year = String.valueOf(yearOfBirth).substring(2);
        String pattern = "^" + year + "-" + designation + "-[0-9]{3}$";
        if(!empID.matches(pattern)) {
            throw new InvalidEmployeeCode("Invalid Employee Code");
        }
    }
    static void retire(Employee e) throws InvalidAgeException {
        int current_year = Calendar.getInstance().get(Calendar.YEAR);
        if(current_year - e.yearOfBirth > 65) {
            throw new InvalidAgeException("Age above 65. Must retire now.");
        }
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the employee details: ");
    System.out.print("Enter Name: ");
    String name = sc.next() + sc.nextLine();
    System.out.print("Enter Employee ID: ");
    String empID = sc.next();
}

```

```
System.out.print("Enter Year of Birth: ");
int yearOfBirth = sc.nextInt();
System.out.print("Enter Designation (F - Faculty, S - Staff): ");
char designation = sc.next().charAt(0);

boolean flag=false;
try {
    validate(empID, yearOfBirth, designation);
}
catch(InvalidEmployeeCode err) {
    flag = true;
    System.out.println(err);
}

if(!flag) {
    System.out.println("VALID DETAILS");
    Employee obj = new Employee(name, empID, yearOfBirth, designation);
    flag = false;
    try {
        retire(obj);
    }
    catch(InvalidAgeException err) {
        flag = true;
        System.out.println(err);
        int current_year = Calendar.getInstance().get(Calendar.YEAR);
        int age = current_year - obj.yearOfBirth;
        String[] id = obj.empID.split("-");
        int num = Integer.parseInt(id[2]);
        int bonus = num*age;
        System.out.println("Bonus Awarded is " + bonus);
    }
    if(!flag) {
        obj.display();
    }
    else {
        System.out.println("Retired.");
    }
}
else {
    System.out.println("INVALID DETAILS");
}
```

## OUTPUT:

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac EmpDetails.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java EmpDetails
Enter the employee details:
Enter Name: Rushabh Kela
Enter Employee ID: 81-F-345
Enter Year of Birth: 1981
Enter Designation (F - Faculty, S - Staff): Faculty
VALID DETAILS

Employee Details:
Name: Rushabh Kela
Employee ID: 81-F-345
Year of Birth: 1981
Designation: Faculty
```

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java EmpDetails
Enter the employee details:
Enter Name: Rushabh Kela
Enter Employee ID: 55-S-452
Enter Year of Birth: 1955
Enter Designation (F - Faculty, S - Staff): Staff
VALID DETAILS
InvalidAgeException: Age above 65. Must retire now.
Bonus Awarded is 29832
Retired.
```

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java EmpDetails
Enter the employee details:
Enter Name: Rushabh Kela
Enter Employee ID: 10234
Enter Year of Birth: 2001
Enter Designation (F - Faculty, S - Staff): Faculty
InvalidEmployeeCode: Invalid Employee Code
INVALID DETAILS
```

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java EmpDetails
Enter the employee details:
Enter Name: Rushabh Kela
Enter Employee ID: 01-F-334
Enter Year of Birth: 2001
Enter Designation (F - Faculty, S - Staff): Staff
InvalidEmployeeCode: Invalid Employee Code
INVALID DETAILS
```

## **7. INTER – THREAD COMMUNICATION –**

**COVID cases are increasing day by day. A new hospital has been inaugurated with maximum capacity of 20 beds with all required facilities. Patients are admitted into the hospital subject to availability of beds. Doctors treat the patient and they are discharged after testing COVID negative. If there are no patients, Doctors wait for new patients to arrive. Write a Java program to illustrate the given scenario using multithreading.**

### **CODE :**

```
import java.util.*;  
  
public class Hospital {  
    public static void main(String[] args) throws InterruptedException  
    {  
        final Admit obj = new Admit();  
        Thread t1 = new Thread(new Runnable() {  
            public void run()  
            {  
                try {  
                    obj.admitPatient();  
                }  
                catch (InterruptedException e) {  
                    System.out.println(e);  
                }  
            }  
        });  
  
        Thread t2 = new Thread(new Runnable() {  
            public void run()  
            {  
                try {  
                    obj.dischargePatient();  
                }  
                catch (InterruptedException e) {  
                    System.out.println(e);  
                }  
            }  
        });  
  
        t1.start();  
        t2.start();  
    }  
}
```

```

try {
    t1.join();
    t2.join();
}
catch(InterruptedException e) {
    System.out.println(e);
}

}

public static class Admit {
    Queue<Integer> queue = new PriorityQueue<>();
    int capacity = 20;

    public void admitPatient() throws InterruptedException
    {
        int value = 1;
        while (true) {
            synchronized (this)
            {
                while (queue.size() >= capacity)
                    wait();

                System.out.println("Patient " + value + " admitted.");
                queue.add(value++);
                notify();
                Thread.sleep(1000);
            }
        }
    }
}

public void dischargePatient() throws InterruptedException
{
    while (true) {
        synchronized (this)
        {
            while (queue.isEmpty())
                wait();

            int val = queue.poll();

            System.out.println("Patient " + val + " discharged.");
        }
    }
}

```

```
        notify();
        Thread.sleep(1000);
    }
}
}
}
}
```

## OUTPUT :

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Hospital.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Hospital
Patient 1 admitted.
Patient 2 admitted.
Patient 3 admitted.
Patient 4 admitted.
Patient 5 admitted.
Patient 1 discharged.
Patient 2 discharged.
Patient 3 discharged.
Patient 4 discharged.
Patient 5 discharged.
Patient 6 admitted.
Patient 7 admitted.
Patient 8 admitted.
Patient 9 admitted.
Patient 10 admitted.
Patient 11 admitted.
Patient 12 admitted.
Patient 13 admitted.
Patient 14 admitted.
Patient 15 admitted.
Patient 16 admitted.
Patient 17 admitted.
Patient 18 admitted.
Patient 19 admitted.
Patient 6 discharged.
Patient 7 discharged.
Patient 8 discharged.
Patient 9 discharged.
Patient 10 discharged.
Patient 11 discharged.
Patient 12 discharged.
Patient 13 discharged.
```

## **8. SERIALIZATION –**

**Hostel counselling is going to start at VIT university. Hence all the students will be ranked as per the NCGPA and called for hostel counselling.**

$$NCGPA = \frac{\text{Your CGPA}}{\text{Topper CGPA}} \times 10$$

**Write a Java program to create a class Student with Registration number, name, CGPA and Proctor Name as its data members. Create 'n' objects of this class for all the students in the university. Write these objects into a file. Store the state of objects of this class in a file. Read these objects from the file and determine the NCGPA of each student and rank them from first to last.**

### **CODE :**

```
import java.io.*;
import java.util.*;

class StudentDetails implements Serializable {
    String regNo;
    String name;
    double cgpa;
    String proctor_name;
    double ncgpa;

    StudentDetails(String regNo, String name, double cgpa, String proctor_name) {
        this.regNo = regNo;
        this.name = name;
        this.cgpa = cgpa;
        this.proctor_name = proctor_name;
    }

    void display() {
        System.out.print(this.name + "\t");
        System.out.print(this.regNo + "\t");
        System.out.print(" " + this.cgpa + "\t");
        System.out.print(this.ncgpa + "\t");
        System.out.print(this.proctor_name + "\t");
        System.out.println();
    }
}
```

```
public class Counselling {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n,i,j;
        System.out.println(" -- VIT HOSTEL COUNSELLING -- ");
        System.out.print("Enter the number of students: ");
        n = sc.nextInt();

        StudentDetails[] students;
        students = new StudentDetails[n];
        double topper_cgpa = 0;
        FileOutputStream fos = new FileOutputStream("hostel.ser");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        for(i = 0; i < n; i++)
        {
            System.out.println("\nEnter the student details: ");
            System.out.print("Enter Register Number: ");
            String regNo = sc.next();
            sc.nextLine();
            System.out.print("Enter name: ");
            String name = sc.nextLine();
            System.out.print("Enter CGPA: ");
            double cgpa = sc.nextDouble();
            sc.nextLine();
            System.out.print("Enter proctor name: ");
            String proctor_name = sc.nextLine();

            topper_cgpa = Math.max(topper_cgpa, cgpa);

            students[i] = new StudentDetails(regNo, name, cgpa, proctor_name);
            oos.writeObject(students[i]);
        }
        System.out.println("\nStudent Data Saved.");
        System.out.println("Reading Data from file.");
        System.out.println("Calculating NCGPA of students.");

        FileInputStream fis = new FileInputStream("hostel.ser");
        ObjectInputStream ois = new ObjectInputStream(fis);
        for(i = 0; i < n; i++)
        {
            StudentDetails sd = (StudentDetails)ois.readObject();
            sd.ncgpa = Math.round((sd.cgpa/topper_cgpa)*1000)/100.0;
            students[i] = sd;
        }
    }
}
```

```
}

for (i = 0; i < n-1; i++)
{
    int max_idx = i;
    for (j = i+1; j < n; j++)
        if (students[j].ncgpa > students[max_idx].ncgpa)
            max_idx = j;

    StudentDetails temp = students[max_idx];
    students[max_idx] = students[i];
    students[i] = temp;
}

System.out.println("Rank List of Students: ");
System.out.println("Rank  Student Name  Register Number  CGPA  NCGPA
Proctor Name");
for(i = 0; i < n; i++)
{
    System.out.print((i+1) + "\t");
    students[i].display();
}
System.out.println("\nRank List Published.");
System.out.println("Hostel Rooms Allotted.");
}

}
```

## OUTPUT:

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Counselling.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Counselling
-- VIT HOSTEL COUNSELLING --
Enter the number of students: 3

Enter the student details:
Enter Register Number: 19BDS0055
Enter name: Rushabh Kela
Enter CGPA: 9.93
Enter proctor name: Prof. Raju R L N

Enter the student details:
Enter Register Number: 19BCE0343
Enter name: Abhinav Chawla
Enter CGPA: 9.70
Enter proctor name: Prof. Deepika S

Enter the student details:
Enter Register Number: 19BCE0134
Enter name: Ashutosh
Enter CGPA: 8.31
Enter proctor name: Prof. Rajkumar R

Student Data Saved.
Reading Data from file.
Calculating NCGPA of students.
Rank List of Students:
Rank    Student Name    Register Number    CGPA    NCGPA    Proctor Name
1        Rushabh Kela    19BDS0055        9.93    10.0     Prof. Raju R L N
2        Abhinav Chawla  19BCE0343        9.7     9.77     Prof. Deepika S
3        Ashutosh         19BCE0134        8.31    8.37     Prof. Rajkumar R

Rank List Published.
Hostel Rooms Allotted.
```

9. Assume only a maximum of 3 placement offers can be given to a student through PAT office. Create a hashmap ‘jobOffers’ with ‘n’ key-value pairs where keys are the names of students and values are companies placed in. Create another hashmap ‘packages’ with ‘m’ key-value pairs where keys are the names of companies and value is the CTC offered by it. Populate the hashmaps with appropriate user inputs and write appropriate code to

- Add or remove a student from ‘jobOffers’
- Iterate over the maps and display the key-value pairs stored in them
- For every student name, fetch the offers given to him and display the company of highest offer given to the student

if the elements of ‘jobOffers’ are

Student	Companies
Rushabh	Microsoft, Oracle
John	Google, Facebook, Microsoft
Charles	JP Morgan, Microsoft

and if the elements of ‘packages’ are

Companies	Package Offered
Microsoft	44,00,000
Oracle	25,00,000
Google	78,00,000
Facebook	32,00,000
JP Morgan	15,00,000

For the student “Charles”, the offers are from JP Morgan of 15,00,000 and from Microsoft of 44,00,000. He will choose Microsoft offer.

Also calculate the average CTC of placements

**CODE :**

```
import java.io.*;
import java.util.*;

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

```

Scanner sc = new Scanner(System.in);
int n,i,m;
System.out.println("Placement And Training Office, VIT");
System.out.print("Enter number of students: ");
n = sc.nextInt();

HashMap<String, String[]> jobOffers = new HashMap<String, String[]>();
System.out.println("Enter the Job Offers: ");
for(i = 0; i < n; i++)
{
    System.out.println("Student " + (i+1) + ": ");
    System.out.print("Enter Name: ");
    String name = sc.next() + sc.nextLine();
    System.out.print("Enter the offers given: ");
    String offers = sc.nextLine();

    String[] arr = offers.split(",");
    jobOffers.put(name, arr);
}

System.out.print("\nEnter number of companies: ");
m = sc.nextInt();
HashMap<String, Long> packages = new HashMap<String, Long>();
for(i = 0; i < m; i++)
{
    System.out.print("\nEnter company name: ");
    String company = sc.next() + sc.nextLine();
    System.out.print("Enter package offered: ");
    long packageOffer = sc.nextLong();

    packages.put(company, packageOffer);
}

System.out.println("Data Saved.");

// Add or delete student from 'jobOffers' hashmap
System.out.println("Edit Registered Students : ");
int ch;
do {
    System.out.println("\n1. Add Student");
    System.out.println("2. Remove Student");
    System.out.println("3. Exit");
    System.out.print("Enter your choice: ");
}

```

```

ch = sc.nextInt();
switch(ch)
{
    case 1:
        System.out.print("Enter Name: ");
        String name = sc.next() + sc.nextLine();
        System.out.print("Enter the offers given: ");
        String offers = sc.nextLine();
        String[] arr = offers.split(",");
        jobOffers.put(name, arr);
        System.out.println("Student added.");
        break;

    case 2:
        System.out.print("Enter name of student to remove: ");
        String rem = sc.next() + sc.nextLine();
        jobOffers.remove(rem);
        System.out.println("Student Removed.");
        break;

    case 3:
        System.out.println("Exit\n");
    }

}while(ch!=3);

// Iterating over the hashmaps
System.out.println("FINAL LISTS : ");
System.out.println("\nStudent\t\tCompanies");
Set<Map.Entry<String, String[]>> set1 = jobOffers.entrySet();
set1.forEach((e) -> {
    System.out.println(e.getKey() + "\t\t" + String.join(", ", e.getValue()));
});

System.out.println("\nCompanies\tPackage Offered");
Set<Map.Entry<String, Long>> set2 = packages.entrySet();
set2.forEach((e) -> {
    System.out.println(e.getKey() + "\t" + e.getValue());
});

// Displaying the offers to students along with highest offer
ArrayList<Long> acceptedOffers = new ArrayList<Long>();
System.out.println("\nJob Offers of Students");
set1.forEach((e) -> {

```

```

        System.out.println("\nFor student " + e.getKey() + ", the offers are from ");
        String[] array = e.getValue();
        String accept = array[0];
        for(String s: array) {
            System.out.println(s + " with package of Rs. " + packages.get(s));

            if(packages.get(s) > packages.get(accept)) {
                accept = s;
            }
        }
        System.out.println("Student will choose " + accept + " offer.");
        acceptedOffers.add(packages.get(accept));
    });

Iterator<Long> itr = acceptedOffers.iterator();
long sum = 0;
while(itr.hasNext()) {
    sum += itr.next();
}
double avgCTC = Math.round(((double)sum/n)*100)/100.0;
System.out.println("Average CTC of current year: Rs. " + avgCTC);
}
}

```

## OUTPUT :

```

[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Placements.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Placements
Placement And Training Office, VIT
Enter number of students: 3
Enter the Job Offers:
Student 1:
Enter Name: Rushabh Kela
Enter the offers given: Microsoft,Oracle
Student 2:
Enter Name: Smith Jones
Enter the offers given: Oracle
Student 3:
Enter Name: Charles
Enter the offers given: JP Morgan,Microsoft

```

```
Enter number of companies: 5

Enter company name: Microsoft
Enter package offered: 4400000

Enter company name: Oracle
Enter package offered: 2500000

Enter company name: Google
Enter package offered: 7800000

Enter company name: Facebook
Enter package offered: 3200000

Enter company name: JP Morgan
Enter package offered: 1500000
Data Saved.

Edit Registered Students :

1. Add Student
2. Remove Student
3. Exit
Enter your choice: 2
Enter name of student to remove: Smith Jones
Student Removed.

1. Add Student
2. Remove Student
3. Exit
Enter your choice: 1
Enter Name: John
Enter the offers given: Google,Facebook,Microsoft
Student added.

1. Add Student
2. Remove Student
3. Exit
Enter your choice: 3
Exit

FINAL LISTS :

Student          Companies
Charles          JP Morgan, Microsoft
John            Google, Facebook, Microsoft
Rushabh Kela    Microsoft, Oracle
```

Companies	Package Offered
Google	7800000
Microsoft	4400000
JP Morgan	1500000
Oracle	2500000
Facebook	3200000

### Job Offers of Students

For student Charles, the offers are from  
 JP Morgan with package of Rs. 1500000  
 Microsoft with package of Rs. 4400000  
 Student will choose Microsoft offer.

For student John, the offers are from  
 Google with package of Rs. 7800000  
 Facebook with package of Rs. 3200000  
 Microsoft with package of Rs. 4400000  
 Student will choose Google offer.

For student Rushabh Kela, the offers are from  
 Microsoft with package of Rs. 4400000  
 Oracle with package of Rs. 2500000  
 Student will choose Microsoft offer.  
 Average CTC of current year: Rs. 5533333.33  
 |Rushabhs-MacBook-Pro:TH-DA-Java rushabh\$

## **10. GENERIC CLASSES AND METHODS –**

**Write a Java program which alters the list according to the specific conditions which the array stores. If the array stores strings, reverse all those elements of array that are not palindrome and add a \* to the palindromic strings. If the array stores integers, replace all the prime numbers in the array with its square.**

### **CODE :**

```
import java.io.*;
import java.util.*;
import java.lang.*;

public class Generics {
    public static < E > boolean isPalindrome(E str) {
        int i;
        String s = str.toString();

        StringBuilder sb = new StringBuilder();
        sb.append(s);
        sb.reverse();
        if(sb.toString().equals(s)) {
            return true;
        }
        return false;
    }

    public static < E > boolean isPrime(E str) {
        int i;
        int n = Integer.parseInt(str.toString());
        for(i = 2; i < n; i++) {
            if(n%i == 0) {
                return false;
            }
        }
        return true;
    }

    public static < E > E[] changeArray(E[] arr) {
        int n = arr.length, i;

        if(arr instanceof String[])
            for(i = 0; i < n; i++) {
                if(isPalindrome(arr[i])) {
                    arr[i] = arr[i].concat("*");
                }
            }
        else {
            for(i = 0; i < n; i++) {
                if(isPrime(arr[i])) {
                    arr[i] = arr[i] * arr[i];
                }
            }
        }
        return arr;
    }
}
```

```

{
    System.out.println("\nReversing all the non - palindrome strings in array: ");
    for(i = 0; i < n; i++)
    {
        if(!isPalindrome(arr[i])) {
            StringBuilder sb = new StringBuilder();
            sb.append(arr[i].toString());
            sb.reverse();
            arr[i] = (E)sb.toString();
        }
        else {
            StringBuilder sb = new StringBuilder();
            sb.append(arr[i].toString());
            sb.append("*");
            arr[i] = (E)sb.toString();
        }
    }
    return arr;
}
if(arr instanceof Integer[])
{
    System.out.println("\nReplacing all prime numbers in the array with their
squares: ");
    for(i = 0; i < n; i++)
    {
        if(isPrime(arr[i])) {
            int m = Integer.parseInt(arr[i].toString());
            Integer sq = new Integer((int)Math.pow(m,2));
            arr[i] = (E)sq;
        }
    }
    return arr;
}
return arr;
}
public static void main(String[] args) {
    System.out.println("GENERIC METHODS");

    Integer[] intArray = {20, 11, 78, 23, 89};
    String[] strArray = {"rushabh", "wow", "abba", "kela"};

    System.out.println("Original Integer Array: ");
    for(int i : intArray) {

```

```

        System.out.print(i + " ");
    }
    Integer[] changedIntArray = changeArray(intArray);
    System.out.println("Changed Integer Array: ");
    for(int i : changedIntArray) {
        System.out.print(i + " ");
    }
    System.out.println("\n");

    System.out.println("Original String Array: ");
    for(String str : strArray) {
        System.out.print(str + " ");
    }
    String[] changedStrArray = changeArray(strArray);
    System.out.println("Changed String Array: ");
    for(String str : changedStrArray) {
        System.out.print(str + " ");
    }
    System.out.println("\n");
}
}

```

## OUTPUT:

```

[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Generics.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Generics
GENERIC METHODS
Original Integer Array:
20 11 78 23 89
Replacing all prime numbers in the array with their squares:
Changed Integer Array:
20 121 78 529 7921

Original String Array:
rushabh wow abba kela
Reversing all the non - palindrome strings in array:
Changed String Array:
hbahsur wow* abba* alek

```

## **11.GUI WITH JavaFX –**

**Winter Semester results are going to be released and COE has to calculate the total marks of students in different courses. There are four different types of courses along with weightage is given below :**

- **TH – Theory Only (100%)**
- **LAB – Lab Only (100%)**
- **ETL – Theory (75%) + Lab (25%)**
- **ETLP – Theory (50%) + Lab (25%) + Project (25%)**

**Develop a GUI with menu and menu items corresponding to the four different courses and teacher must be able to enter the marks of**

- **CAT 1, CAT 2, Quiz 1, Quiz 2, DA 1, FAT in theory component**
- **3 Lab assessment, 2 LabCATs, LabFAT mark in lab component**
- **Review 1, Review 2, Review 3 marks in project component**

**Use buttons, text fields and event handlers.**

## **CODE :**

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.scene.text.*;
import javafx.event.*;
import javafx.geometry.*;
import javafx.scene.control.*;
import javafx.stage.Stage;
import javafx.scene.image.*;
import java.io.*;

public class App extends Application {
    TextField cat1_marks;
    TextField cat2_marks;
    TextField quiz1_marks;
    TextField quiz2_marks;
    TextField da_marks;
    TextField fat_marks;
    TextField labcat1_marks;
    TextField labcat2_marks;
    TextField ass1_marks;
    TextField ass2_marks;
```

```
TextField ass3_marks;
TextField labfat_marks;
TextField review1_marks;
TextField review2_marks;
TextField review3_marks;
GridPane theory() {
    GridPane grid = new GridPane();
    grid.setAlignment(Pos.CENTER);
    grid.setHgap(10);
    grid.setVgap(10);
    grid.setPadding(new Insets(25, 25, 25, 25));

    Label cat1 = new Label("CAT1 : ");
    grid.add(cat1, 0, 7);
    cat1_marks = new TextField();
    grid.add(cat1_marks, 1, 7);

    Label cat2 = new Label("CAT2 : ");
    grid.add(cat2, 0, 8);
    cat2_marks = new TextField();
    grid.add(cat2_marks, 1, 8);

    Label quiz1 = new Label("Quiz 1 : ");
    grid.add(quiz1, 0, 9);
    quiz1_marks = new TextField();
    grid.add(quiz1_marks, 1, 9);

    Label quiz2 = new Label("Quiz 2 : ");
    grid.add(quiz2, 0, 10);
    quiz2_marks = new TextField();
    grid.add(quiz2_marks, 1, 10);

    Label da = new Label("Digital Assignment : ");
    grid.add(da, 0, 11);
    da_marks = new TextField();
    grid.add(da_marks, 1, 11);

    Label fat = new Label("FAT : ");
    grid.add(fat, 0, 12);
    fat_marks = new TextField();
    grid.add(fat_marks, 1, 12);
    return grid;
}
```

```
GridPane lab() {
    GridPane grid = new GridPane();
    grid.setAlignment(Pos.CENTER);
    grid.setHgap(10);
    grid.setVgap(10);
    grid.setPadding(new Insets(25, 25, 25, 25));

    Label labcat1 = new Label("LabCAT1 : ");
    grid.add(labcat1, 0, 7);
    labcat1_marks = new TextField();
    grid.add(labcat1_marks, 1, 7);

    Label labcat2 = new Label("LabCAT2 : ");
    grid.add(labcat2, 0, 8);
    labcat2_marks = new TextField();
    grid.add(labcat2_marks, 1, 8);

    Label ass1 = new Label("Assessment 1 : ");
    grid.add(ass1, 0, 9);
    ass1_marks = new TextField();
    grid.add(ass1_marks, 1, 9);

    Label ass2 = new Label("Assessment 2 : ");
    grid.add(ass2, 0, 10);
    ass2_marks = new TextField();
    grid.add(ass2_marks, 1, 10);

    Label ass3 = new Label("Assessment 3 : ");
    grid.add(ass3, 0, 11);
    ass3_marks = new TextField();
    grid.add(ass3_marks, 1, 11);

    Label labfat = new Label("LabFAT : ");
    grid.add(labfat, 0, 12);
    labfat_marks = new TextField();
    grid.add(labfat_marks, 1, 12);

    return grid;
}

GridPane project() {
    GridPane grid = new GridPane();
```

```

grid.setAlignment(Pos.CENTER);
grid.setHgap(10);
grid.setVgap(10);
grid.setPadding(new Insets(25, 25, 25, 25));

Label review1 = new Label("Review 1 : ");
grid.add(review1, 0, 7);
review1_marks = new TextField();
grid.add(review1_marks, 1, 7);

Label review2 = new Label("Review 2 : ");
grid.add(review2, 0, 8);
review2_marks = new TextField();
grid.add(review2_marks, 1, 8);

Label review3 = new Label("Review 3 : ");
grid.add(review3, 0, 9);
review3_marks = new TextField();
grid.add(review3_marks, 1, 9);

return grid;
}

public void start(Stage s) throws Exception
{
    s.setTitle("Vellore Institute of Technology, Vellore");
    Menu home = new Menu("Home");
    Menu courses = new Menu("Course Types");

    MenuItem home1 = new MenuItem("Faculty Info");
    MenuItem home2 = new MenuItem("Grading Schemes");

    MenuItem course1 = new MenuItem("TH - Theory Only");
    MenuItem course2 = new MenuItem("LO - Lab Only");
    MenuItem course3 = new MenuItem("ETL - Embedded Theory and Lab");
    MenuItem course4 = new MenuItem("ETLP - Embedded Theory, Lab and
Project");

    home.getItems().addAll(home1, home2);
    courses.getItems().addAll(course1, course2, course3, course4);

    MenuBar mb = new MenuBar();
    mb.getMenus().addAll(home, courses);
}

```

```

FileInputStream input = new FileInputStream("vit.png");
Image image = new Image(input);
ImageView imageView = new ImageView(image);
imageView.setFitHeight(200);
imageView.setFitWidth(750);

Label l1 = new Label("", imageView);
Label l2 = new Label("Welcome to Vellore Institute of Technology, Vellore");
l2.setWrapText(true);
l2.setPadding(new Insets(20));
l2.setFont(Font.font ("Arial", 35));

EventHandler<ActionEvent> event1 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        String name = "Faculty Name: Jaisankar N\n";
        String empID = "Employee ID: 10247\n";
        String cabin = "Cabin Number: SJT-323-A\n";
        String designation = "Designation: Higher Academic Grade\n";
        String school = "School: SCOPE\n";
        String email = "Email: njaisankar@vit.ac.in";
        l2.setText("Faculty Information\n" + name + empID + cabin + designation +
school + email);
    }
};

EventHandler<ActionEvent> event2 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        String th = "\n\t1. TH – Theory Only (100%)\n";
        String lo = "\t2. LAB – Lab Only (100%)\n";
        String etl = "\t3. ETL – Theory (75%) + Lab (25%)\n";
        String etlp = "\t4. Theory (50%) + Lab (25%) + Project (25%)";
        l2.setText("Types of courses along with weightage is given below :" + th + lo
+ etl + etlp);
    }
};

EventHandler<ActionEvent> event3 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        GridPane grid = theory();
    }
};

```

```

VBox vb = new VBox(mb, grid);
Scene scene = new Scene(vb, 1000, 600);
Text scenetitle = new Text("Theory Only Course - TH\nMark Entry: ");
Label note = new Label("\n\n\nTheory Component (100%)\nMaximum
marks for CATs are 30, Quizzes and Assignments are 10, and FAT is 60.");
scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
note.setWrapText(true);
grid.add(scenetitle, 0, 0, 2, 1);
grid.add(note, 0, 0, 2, 2);

Label totals = new Label("TOTAL MARKS = 0");
totals.setFont(Font.font("Tahoma", FontWeight.BOLD, 25));
grid.add(totals, 0, 14, 2, 1);

EventHandler<ActionEvent> event1 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        double total = Double.parseDouble(cat1_marks.getText())/2 +
                      Double.parseDouble(cat2_marks.getText())/2 +
                      Double.parseDouble(quiz1_marks.getText()) +
                      Double.parseDouble(quiz2_marks.getText()) +
                      Double.parseDouble(da_marks.getText()) +
                      ((Double.parseDouble(fat_marks.getText())/60)*40);

        totals.setText("TOTAL MARKS = " + String.format("%.2f",total));
    }
};

Button button = new Button("Calculate Total");
button.setOnAction(event1);
grid.add(button, 1, 13);
s.setScene(scene);
};

EventHandler<ActionEvent> event4 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        GridPane grid = lab();
        VBox vb = new VBox(mb, grid);
        Scene scene = new Scene(vb, 1000, 600);
        Text scenetitle = new Text("Lab Only Course - LO\nMark Entry: ");

```

```

Label note = new Label("\n\n\nLab Component (100%)\nMaximum marks
for LabCATs are 15, Assessments are 10, and LabFAT is 50.");
scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
note.setWrapText(true);
grid.add(scenetitle, 0, 0, 2, 1);
grid.add(note, 0, 0, 2, 2);

Label totals = new Label("TOTAL MARKS = 0");
totals.setFont(Font.font("Tahoma", FontWeight.BOLD, 25));
grid.add(totals, 0, 14, 2, 1);

EventHandler<ActionEvent> event1 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        double total = Double.parseDouble(labcat1_marks.getText()) +
                      Double.parseDouble(labcat2_marks.getText()) +
                      Double.parseDouble(ass1_marks.getText()) +
                      Double.parseDouble(ass2_marks.getText()) +
                      Double.parseDouble(ass3_marks.getText()) +
                      ((Double.parseDouble(labfat_marks.getText())/50)*40);

        totals.setText("TOTAL MARKS = " + String.format("%.2f",total));
    }
};

Button button = new Button("Calculate Total");
button.setOnAction(event1);
grid.add(button, 1, 13);
s.setScene(scene);
}

};

EventHandler<ActionEvent> event5 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        Text scenetitle = new Text("\n Embedded Theory and Lab Course - ETL\n
Mark Entry: ");
        Label note = new Label("Theory Component(75%) - Lab Component (25%)");
        Label th_note = new Label("Maximum marks for CATs are 30, Quizzes and
Assignments are 10, and FAT is 60.");
        Label lab_note = new Label("Maximum marks for LabCATs are 15,
Assessments are 10, and LabFAT is 50.");
        scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));

```

```

note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
note.setPadding(new Insets(20,0,0,20));
th_note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
th_note.setPadding(new Insets(0,0,0,20));
lab_note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
lab_note.setPadding(new Insets(0,0,0,20));

Label totals = new Label("TOTAL MARKS = 0");
totals.setFont(Font.font("Tahoma", FontWeight.BOLD, 25));
totals.setPadding(new Insets(20));

EventHandler<ActionEvent> event1 = new EventHandler<ActionEvent>() {
    public void handle(ActionEvent e)
    {
        double total = ((Double.parseDouble(cat1_marks.getText())/2 +
Double.parseDouble(cat2_marks.getText())/2 +
Double.parseDouble(quiz1_marks.getText()) +
Double.parseDouble(quiz2_marks.getText()) +
Double.parseDouble(da_marks.getText()) +
((Double.parseDouble(fat_marks.getText())/60)*40)*0.75) +
((Double.parseDouble(labcat1_marks.getText()) +
Double.parseDouble(labcat2_marks.getText()) +
Double.parseDouble(ass1_marks.getText()) +
Double.parseDouble(ass2_marks.getText()) +
Double.parseDouble(ass3_marks.getText()) +
((Double.parseDouble(labfat_marks.getText())/50)*40))*0.25);

        totals.setText("TOTAL MARKS = " + String.format("%.2f",total));
    }
};

Button button = new Button("Calculate Total");
button.setOnAction(event1);

HBox hbox = new HBox(theory(), lab());
VBox vb = new VBox(mb, scenetitle, note, th_note, lab_note, hbox, button,
totals);
VBox.setMargin(button, new Insets(0,0,0,20));
Scene scene = new Scene(vb, 1000, 600);
s.setScene(scene);
};

};

EventHandler<ActionEvent> event6 = new EventHandler<ActionEvent>() {

```

```

public void handle(ActionEvent e)
{
    Text scenetitle = new Text("\n Embedded Theory, Lab and Project Course -\n ETLP\n Mark Entry:");
    Label note = new Label("Theory Component(50%) - Lab Component (25%) -\n Project Component (25%)");
    Label th_note = new Label("Maximum marks for CATs are 30, Quizzes and\n Assignments are 10, and FAT is 60.");
    Label lab_note = new Label("Maximum marks for LabCATs are 15,\n Assessments are 10, and LabFAT is 50.");
    Label j_note = new Label("Maximum marks for Review 1 is 20, Review 2 is\n 30 and Review 3 is 50.");
    scenetitle.setFont(Font.font("Tahoma", FontWeight.NORMAL, 20));
    note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
    note.setPadding(new Insets(20,0,0,20));
    th_note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
    th_note.setPadding(new Insets(0,0,0,20));
    lab_note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
    lab_note.setPadding(new Insets(0,0,0,20));
    j_note.setFont(Font.font("Tahoma", FontWeight.NORMAL, 15));
    j_note.setPadding(new Insets(0,0,0,20));

    Label totals = new Label("TOTAL MARKS = 0");
    totals.setFont(Font.font("Tahoma", FontWeight.BOLD, 25));
    totals.setPadding(new Insets(20));

    EventHandler<ActionEvent> event1 = new EventHandler<ActionEvent>() {
        public void handle(ActionEvent e)
        {
            double total = ((Double.parseDouble(cat1_marks.getText())/2 +
Double.parseDouble(cat2_marks.getText())/2 +
Double.parseDouble(quiz1_marks.getText()) +
Double.parseDouble(quiz2_marks.getText()) +
Double.parseDouble(da_marks.getText()) +
((Double.parseDouble(fat_marks.getText())/60)*40))*0.5) +
                ((Double.parseDouble(labcat1_marks.getText()) +
Double.parseDouble(labcat2_marks.getText()) +
Double.parseDouble(ass1_marks.getText()) +
Double.parseDouble(ass2_marks.getText()) +
Double.parseDouble(ass3_marks.getText()) +
((Double.parseDouble(labfat_marks.getText())/50)*40))*0.25) +

```

```

((Double.parseDouble(review1_marks.getText()) +
Double.parseDouble(review2_marks.getText()) +
Double.parseDouble(review3_marks.getText()))*0.25);

        totals.setText("TOTAL MARKS = " + String.format("%.2f",total));
    }
};

Button button = new Button("Calculate Total");
button.setOnAction(event1);

HBox hbox = new HBox(theory(), lab(), project());
VBox vb = new VBox(mb,scenetitle, note, th_note, lab_note, hbox, button,
totals);
VBox.setMargin(button, new Insets(0,0,0,20));
Scene scene = new Scene(vb, 1000, 600);
s.setScene(scene);
}

};

home1.setOnAction(event1);
home2.setOnAction(event2);

course1.setOnAction(event3);
course2.setOnAction(event4);
course3.setOnAction(event5);
course4.setOnAction(event6);

VBox vb = new VBox(mb, l1, l2);
Scene sc = new Scene(vb, 1000, 600);
s.setScene(sc);
s.show();
}

public static void main(String args[])
{
    launch(args);
}
}
```

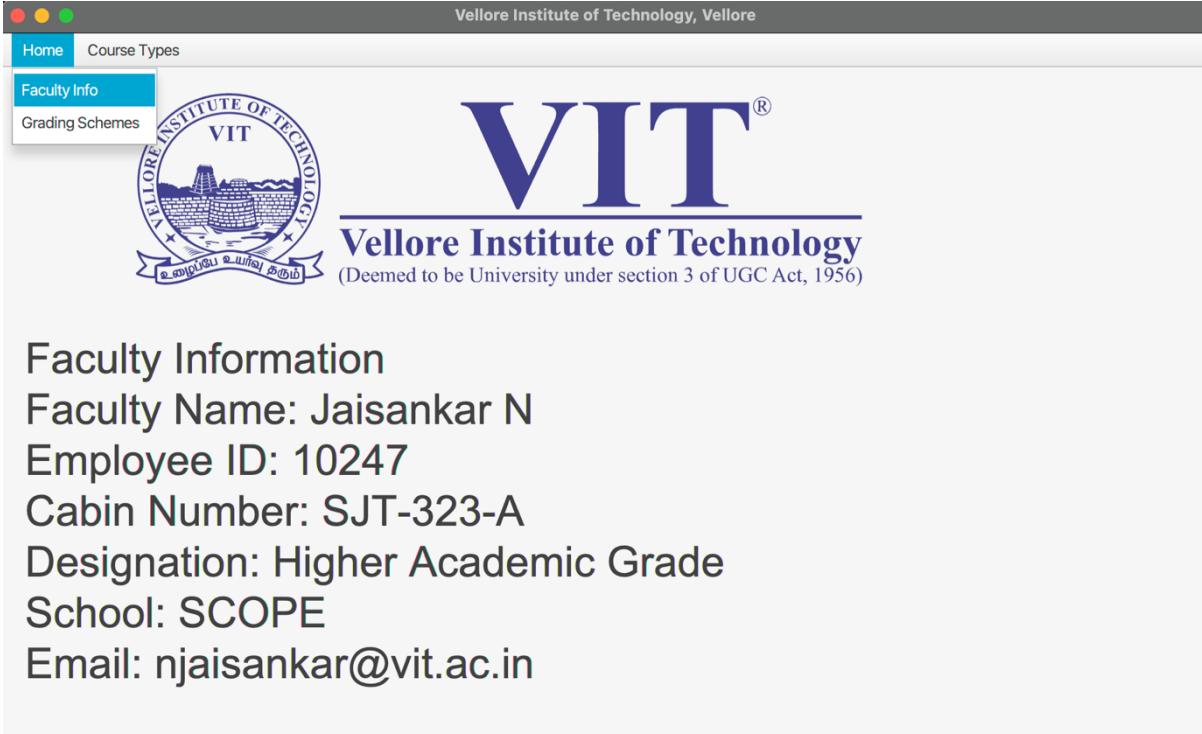
## **OUTPUT :**

### **■ HOME PAGE :**

The screenshot shows the official website of Vellore Institute of Technology. At the top, there is a navigation bar with three dots (red, yellow, green) on the left, followed by the text "Vellore Institute of Technology, Vellore". Below the navigation bar, there are two tabs: "Home" (which is highlighted in blue) and "Course Types". Under the "Home" tab, there are links to "Faculty Info" and "Grading Schemes". To the right of these links is the VIT logo, which is a circular emblem featuring a temple gopuram and the text "VELLORE INSTITUTE OF TECHNOLOGY" around the top and "VIT" in the center. Below the logo, the text "Vellore Institute of Technology" is written in a large, bold, blue font, with "(Deemed to be University under section 3 of UGC Act, 1956)" in smaller text underneath. The main content area features a large "VIT" logo and the text "Welcome to Vellore Institute of Technology, Vellore".

This screenshot is identical to the one above, showing the VIT home page. It features the same navigation bar, tabs, logo, and welcome message. The only difference is the content in the "Course Types" dropdown menu, which now lists four options: "TH - Theory Only", "LO - Lab Only", "ETL - Embedded Theory and Lab", and "ETLP - Embedded Theory, Lab and Project".

## ■ FACULTY INFO :

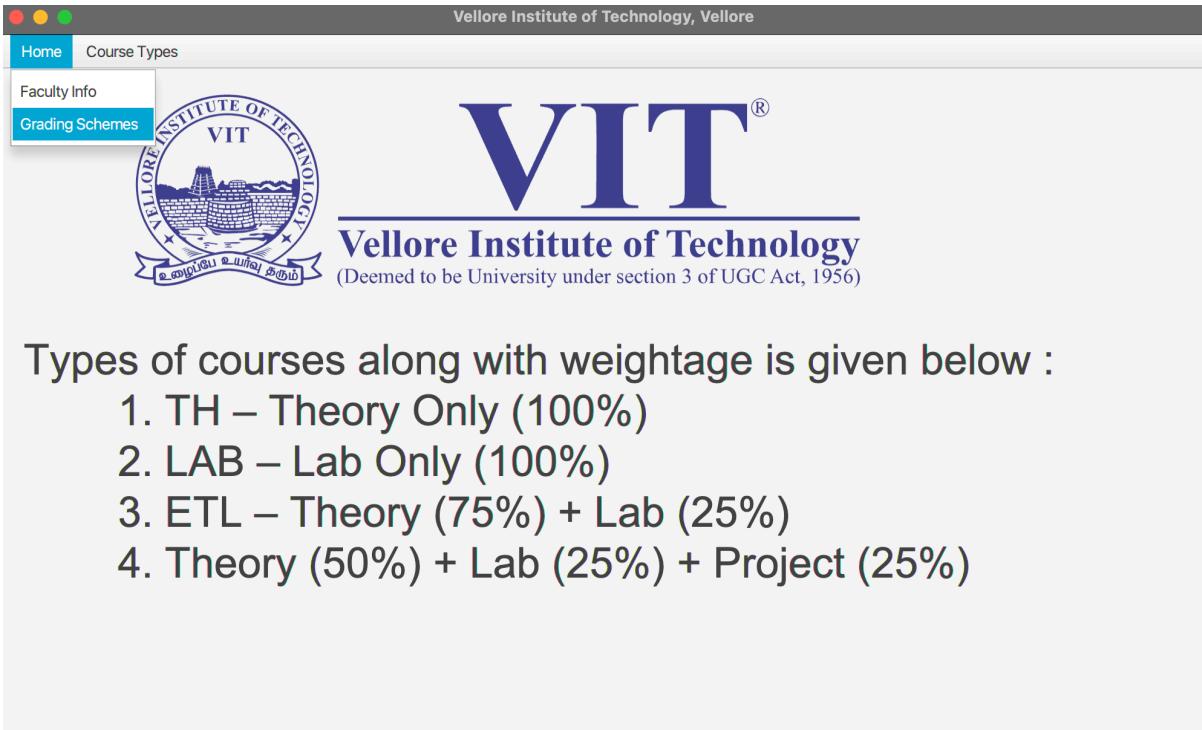


The screenshot shows the Vellore Institute of Technology (VIT) website. The top navigation bar includes links for Home, Course Types, Faculty Info (which is selected), and Grading Schemes. The main content area features the VIT logo, the text "VIT®", "Vellore Institute of Technology", and the subtitle "(Deemed to be University under section 3 of UGC Act, 1956)". Below this, faculty information is displayed for Jaisankar N.

**Faculty Information**

Faculty Name: Jaisankar N  
Employee ID: 10247  
Cabin Number: SJT-323-A  
Designation: Higher Academic Grade  
School: SCOPE  
Email: njaisankar@vit.ac.in

## ■ GRADING SCHEME :



The screenshot shows the Vellore Institute of Technology (VIT) website. The top navigation bar includes links for Home, Course Types, Faculty Info, and Grading Schemes (which is selected). The main content area features the VIT logo, the text "VIT®", "Vellore Institute of Technology", and the subtitle "(Deemed to be University under section 3 of UGC Act, 1956)". Below this, the grading scheme is described.

Types of courses along with weightage is given below :

1. TH – Theory Only (100%)
2. LAB – Lab Only (100%)
3. ETL – Theory (75%) + Lab (25%)
4. Theory (50%) + Lab (25%) + Project (25%)

- MARK ENTRY FOR THEORY ONLY – COURSE  
( *Menu → Course Types → TH – Theory Only* )

The screenshot shows a web application for marking theory-only courses at Vellore Institute of Technology. The header includes the university's name and navigation links for Home and Course Types. The main content area is titled "Theory Only Course - TH" and "Mark Entry:". It provides a breakdown of the theory component marks: CATs (30), Quizzes (10), Assignments (10), and FAT (60). Below this, individual marks are entered for each category: CAT1 (25), CAT2 (21), Quiz 1 (9.5), Quiz 2 (10), Digital Assignment (10), and FAT (47). A "Calculate Total" button is present, and the total calculated is **TOTAL MARKS = 83.83**.

CAT1:	25
CAT2:	21
Quiz 1:	9.5
Quiz 2 :	10
Digital Assignment :	10
FAT:	47

**Calculate Total**

**TOTAL MARKS = 83.83**

- MARK ENTRY FOR LAB ONLY – COURSE  
( *Menu → Course Types → LO – Lab Only* )

The screenshot shows a web application for marking lab-only courses. At the top, there's a header bar with the Vellore Institute of Technology logo and navigation links for Home and Course Types. Below the header, the page title is "Lab Only Course - LO Mark Entry:". A note below the title states: "Lab Component (100%) Maximum marks for LabCATs are 15, Assessments are 10, and LabFAT is 50." The form contains fields for entering marks for LabCAT1 (14), LabCAT2 (13), Assessment 1 (10), Assessment 2 (10), Assessment 3 (9), and LabFAT (45). A "Calculate Total" button is present, and the total marks calculated are displayed as **TOTAL MARKS = 92.00**.

Vellore Institute of Technology, Vellore

Home Course Types

Lab Only Course - LO  
Mark Entry:

Lab Component (100%)  
Maximum marks for LabCATs are 15, Assessments are 10, and LabFAT is 50.

LabCAT1: 14

LabCAT2: 13

Assessment 1: 10

Assessment 2: 10

Assessment 3: 9

LabFAT: 45

Calculate Total

**TOTAL MARKS = 92.00**

- MARK ENTRY FOR EMBEDDED THEORY AND LAB – COURSE  
( *Menu → Course Types → ETL – Embedded Theory and Lab* )

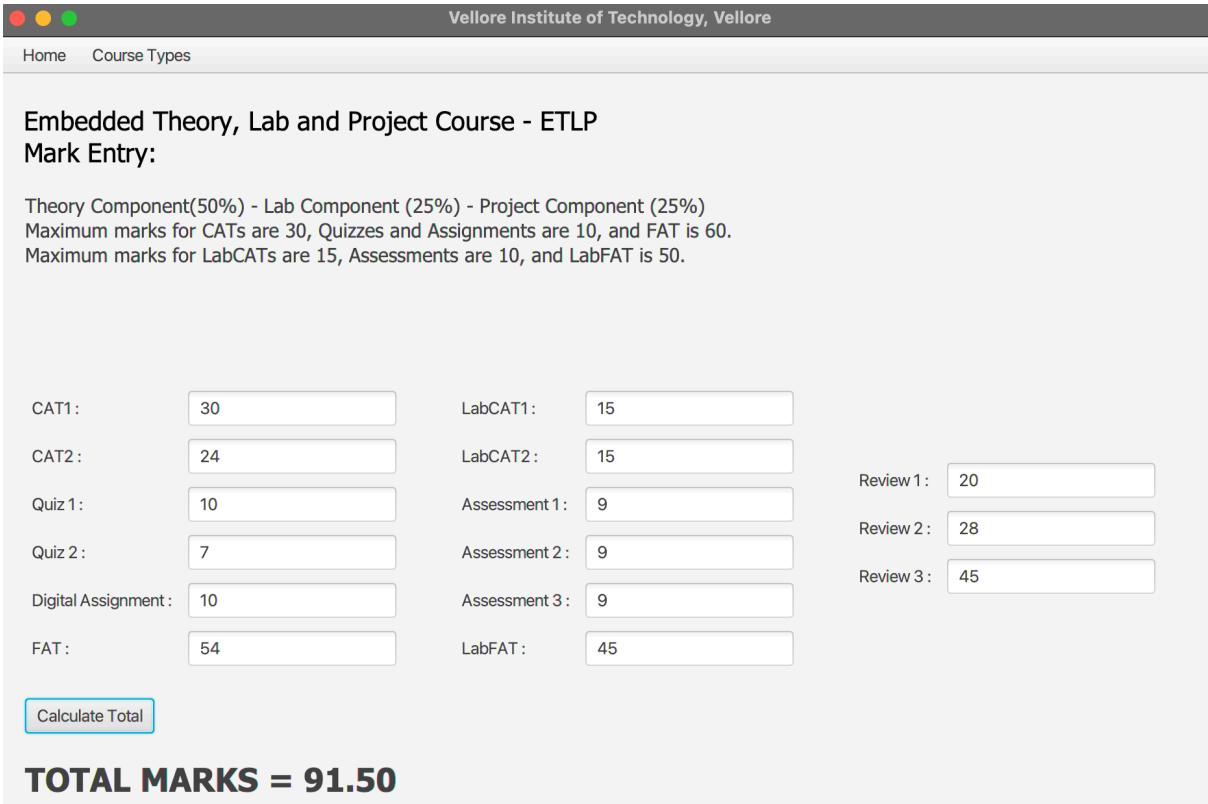
The screenshot shows a web application for marking entries. At the top, there is a header bar with three colored dots (red, yellow, green) and the text "Vellore Institute of Technology, Vellore". Below the header, a navigation bar includes links for "Home" and "Course Types". The main content area is titled "Embedded Theory and Lab Course - ETL" and "Mark Entry:". It contains descriptive text about the component weights and maximum marks for various assessments. Below this, there is a table with input fields for each assessment type, followed by a "Calculate Total" button and a large bolded "TOTAL MARKS = 94.95" at the bottom.

CAT1 :	30	LabCAT1 :	15
CAT2 :	28	LabCAT2 :	15
Quiz 1 :	10	Assessment 1 :	10
Quiz 2 :	6	Assessment 2 :	10
Digital Assignment :	10	Assessment 3 :	10
FAT :	59	LabFAT :	46

**Calculate Total**

**TOTAL MARKS = 94.95**

- **MARK ENTRY FOR EMBEDDED THEORY, LAB AND PROJECT – COURSE**  
**( *Menu → Course Types → ETLP – Embedded Theory, Lab and Project* )**



The screenshot shows a web-based mark entry system for the Embedded Theory, Lab and Project Course - ETLP at Vellore Institute of Technology. The interface includes a header with the college logo and name, a navigation bar with Home and Course Types links, and a main content area for entering marks for various assessment components.

**Assessment Components and Marks:**

CAT1:	30	LabCAT1:	15
CAT2 :	24	LabCAT2 :	15
Quiz 1:	10	Assessment 1:	9
Quiz 2 :	7	Assessment 2:	9
Digital Assignment :	10	Assessment 3:	9
FAT :	54	LabFAT :	45

**Review Marks:**

Review 1:	20
Review 2 :	28
Review 3 :	45

**Total Marks:** **TOTAL MARKS = 91.50**

## **12.POLYMORPHISM –**

**Create an abstract class Hospital which contains the following for a hospital : Name, Type(Government, Private), Code, Email. Then implement an abstract method which determines the amount to be paid by a patient to the hospital which varies between government and private hospitals. For Government hospital the ward cost is flat 5000 Rs and treatment cost is flat 2000 Rs. For private hospitals, ward cost is 500 Rs/hour and treatment cost varies according to treatment (take input from user). Create two child classes GovernmentHospital and PrivateHospital. Write a Java program by creating objects of the two child classes and display the bills details accordingly.**

### **CODE :**

```
import java.util.*;  
  
abstract class Hospitals {  
    String name;  
    String type;  
    int code;  
    String email;  
    String patient_name;  
    abstract int amount();  
    abstract void display();  
  
    Hospitals(String name, String type, int code, String email, String patient_name)  
    {  
        this.name = name;  
        this.type = type;  
        this.code = code;  
        this.email = email;  
        this.patient_name = patient_name;  
    }  
}  
  
class PrivateHospital extends Hospitals {  
    public PrivateHospital(String name, String type, int code, String email, String  
    patient_name) {  
        super(name,type,code,email,patient_name);  
    }  
    public int amount() {
```

```

Scanner sc = new Scanner(System.in);
System.out.println("Treatment Details:");
System.out.print("Enter number of hours for treatment : ");
int hr = sc.nextInt();
System.out.print("Enter the treatment cost : ");
int n = sc.nextInt();

int total_cost = 500*hr + n;
return total_cost;
}
public void display() {
    System.out.println("\nPRIVATE HOSPITAL BILL");
    System.out.println("Hospital Name: " + this.name);
    System.out.println("Hospital Code: " + this.code);
    System.out.println("Email: " + this.email);
    System.out.println("Patient Name: " + this.patient_name);
    System.out.println("Amount to be paid by the patient: Rs. " + this.amount());
}
}

class GovernmentHospital extends Hospitals {
    public GovernmentHospital(String name, String type, int code, String email, String
patient_name) {
        super(name,type,code,email,patient_name);
    }
    public int amount() {
        int treatment = 2000;
        int room = 5000;
        int total_cost = treatment + room;
        return total_cost;
    }
    public void display() {
        System.out.println("\nGOVERNMENT HOSPITAL BILL");
        System.out.println("Hospital Name: " + this.name);
        System.out.println("Hospital Code: " + this.code);
        System.out.println("Email: " + this.email);
        System.out.println("Patient Name: " + this.patient_name);
        System.out.println("Amount to be paid by the patient: Rs. " + this.amount());
    }
}

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

```

```
System.out.println("HOSPITAL BILLS");
Scanner sc = new Scanner(System.in);
System.out.print("Enter hospital name: ");
String name = sc.next() + sc.nextLine();
System.out.print("Enter hospital code: ");
int code = sc.nextInt();
System.out.print("Enter hospital type (Private/Government): ");
String type = sc.next();
System.out.print("Enter hospital email: ");
String email = sc.next();
sc.nextLine();
System.out.print("Enter patient name: ");
String patient_name = sc.nextLine();

if(type.equals("Private"))
{
    Hospitals obj = new PrivateHospital(name, type, code, email, patient_name);
    obj.display();
}
else
{
    Hospitals obj = new GovernmentHospital(name, type, code, email,
patient_name);
    obj.display();
}
```

## **OUTPUT:**

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac HospitalBills.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java HospitalBills
HOSPITAL BILLS
Enter hospital name: Wockhardt Hospitals
Enter hospital code: 1992
Enter hospital type (Private/Government): Private
Enter hospital email: wockhardt.healthcare@gmail.com
Enter patient name: Rushabh Kela

PRIVATE HOSPITAL BILL
Hospital Name: Wockhardt Hospitals
Hospital Code: 1992
Email: wockhardt.healthcare@gmail.com
Patient Name: Rushabh Kela
Treatment Details:
Enter number of hours for treatment : 20
Enter the treatment cost : 4500
Amount to be paid by the patient: Rs. 14500
```

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java HospitalBills
HOSPITAL BILLS
Enter hospital name: City Hospital
Enter hospital code: 3421
Enter hospital type (Private/Government): Government
Enter hospital email: city.hospital@gov.in
Enter patient name: Rushabh Kela

GOVERNMENT HOSPITAL BILL
Hospital Name: City Hospital
Hospital Code: 3421
Email: city.hospital@gov.in
Patient Name: Rushabh Kela
Amount to be paid by the patient: Rs. 7000
```

### **13. FILES AND STACK OPERATION TO EVALUATE EXPRESSION –**

**Write a program that reads an expression from the file and split the operands and operators into the two separate files. Count the number of operands and operators and write it in the respective file in which they have been split and evaluate the expression.**

#### **CODE :**

```
import java.io.*;
import java.util.*;

public class Arithmetic {
    public static boolean hasPrecedence(char op1, char op2)
    {
        if (op2 == '(' || op2 == ')')
            return false;
        if ((op1 == '*' || op1 == '/') && (op2 == '+' || op2 == '-'))
            return false;
        else
            return true;
    }
    public static double applyOp(char op,double b, double a)
    {
        switch (op)
        {
            case '+':
                return a + b;
            case '-':
                return a - b;
            case '*':
                return a * b;
            case '/':
                if (b == 0)
                    throw new UnsupportedOperationException("Cannot divide by zero");
                return a / b;
        }
        return 0;
    }
    public static void main(String[] args) throws IOException {
        FileReader fr = new FileReader("expression.txt");
        BufferedReader br = new BufferedReader(fr);

        FileWriter op1 = new FileWriter("operators.txt");
        PrintWriter pw1 = new PrintWriter(op1);
```

```

FileWriter op2 = new FileWriter("operands.txt");
PrintWriter pw2 = new PrintWriter(op2);
int cnt = 1;
String line = br.readLine();
while(line!=null)
{
    pw1.println("EXPRESSION " + (cnt));
    pw2.println("EXPRESSION " + (cnt++));
    String s = "";
    int operators = 0;
    int operands = 0;
    String exp = "";
    int i;
    for(i = 0; i < line.length(); i++) {
        char k = line.charAt(i);
        exp += k;
        if(Character.isDigit(k)) {
            s += k;
        }
        else {
            pw1.print(s + " ");
            operators++;
            s = "";
            pw2.print(k + " ");
            operands++;
        }
    }
    pw1.print(s);
    operators++;
    pw1.println("\nCount of operators: " + operators + "\n");
    pw2.println("\nCount of operands: " + operands + "\n");

    char[] tokens = exp.toCharArray();
    Stack<Double> values = new Stack<Double>();
    Stack<Character> ops = new Stack<Character>();

    for (i = 0; i < tokens.length; i++)
    {
        if (tokens[i] == ' ')
            continue;
        if (tokens[i] >= '0' && tokens[i] <= '9')
        {
            StringBuffer sbuf = new StringBuffer();

```

```

        while (i < tokens.length && tokens[i] >= '0' && tokens[i] <= '9')
            sbuf.append(tokens[i++]);
        values.push(Double.parseDouble(sbuf.toString())));
        i--;
    }
    else if (tokens[i] == '(')
        ops.push(tokens[i]);
    else if (tokens[i] == ')')
    {
        while (ops.peek() != '(')
            values.push(applyOp(ops.pop(), values.pop(), values.pop()));
        ops.pop();
    }
    else if (tokens[i] == '+' || tokens[i] == '-' || tokens[i] == '*' || tokens[i] == '/')
    {
        while (!ops.empty() && hasPrecedence(tokens[i], ops.peek()))
            values.push(applyOp(ops.pop(), values.pop(), values.pop()));
        ops.push(tokens[i]);
    }
}
while (!ops.empty())
    values.push(applyOp(ops.pop(), values.pop(), values.pop()));

System.out.println("\nEvaluated Expression " + (cnt-1) + " : " + values.pop());
line = br.readLine();
}
pw1.flush();
pw1.close();
pw2.flush();
pw2.close();
}
}

```

## OUTPUT :

- ***expression.txt* file :**

≡ expression.txt ×

≡ expression.txt

1 5+4-10/4\*3

2 6/10\*8+3

- **CODE OUTPUT :**

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Arithmetic.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Arithmetic

Evaluated Expression 1 : 1.5

Evaluated Expression 2 : 7.8
```

- ***operands.txt* file :**

```
≡ operands.txt ×

≡ operands.txt
1      EXPRESSION 1
2      + - / *
3      Count of operands: 4
4
5      EXPRESSION 2
6      / * +
7      Count of operands: 3
```

- ***operators.txt* file :**

```
≡ operators.txt ×

≡ operators.txt
1      EXPRESSION 1
2      5 4 10 4 3
3      Count of operators: 5
4
5      EXPRESSION 2
6      6 10 8 3
7      Count of operators: 4
```

#### **14. SYNCHRONISATION –**

**VIT academics department has decided to change the type of JAVA PROGRAMMING course based on student demands. The options given are to convert the course into “Lab – only”, “Embedded Theory, Lab and Project” or “Embedded Lab and Project”. You have been given the task to count the student votes for each type. With the total students of 2000 people in Computer Science branch, simulate the vote casting by generating 2000 random numbers (1 for lab - only, 2 for theory – lab – project and 3 for lab – project) and store them in an array. Create four threads to equally share the task of counting the number of votes. Use synchronization to update the three count variables. The main thread should receive the final count for all three types and hence decide the new type of the course based on the values received.**

#### **CODE :**

```
import java.lang.*;
import java.util.*;

class Count
{
    int type1;
    int type2;
    int type3;
    int arr[]=new int[2000];
    static int i=0;
    static int k = 0;
    Count(int arr[])
    {
        type1=0;
        type2=0;
        type3=0;
        this.arr=arr;
    }
    void types_count()
    {
        while(i<arr.length)
        {
            synchronized(this)
            {

```

```

        if(arr[i]==1)
            type1++;
        else if(arr[i]==2)
            type2++;
        else
            type3++;

        i++;
    }
    try {
        Thread.sleep(300);
    }
    catch(Exception e) {}
}
}

int[] fxn() {
    int array[]=new int[3];
    array[0]=type1;
    array[1]=type2;
    array[2]=type3;
    System.out.println("\nTotal votes of type 1 - Lab Only : "+array[0]);
    System.out.println("Total types of type 2 - Embedded Theory, Lab, Project :
"+array[1]);
    System.out.println("Total types of type 3 - Lab and Project : "+array[2]+"\n");
    return array;
}
}

class MyRunnable implements Runnable
{
    Count c;
    MyRunnable(Count temp) {
        c=temp;
    }
    public void run() {
        c.types_count();
    }
}

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

```

```

int array[] = new int[2000];
System.out.println(" -- COURSE TYPE CHANGE FOR CSE1007 - JAVA
PROGRAMMING --");
System.out.println("Counting started..");
for(int i=0;i<2000;i++)
{
    array[i]=(int)(Math.random()*3+1);

}
Count obj=new Count(array);
MyRunnable r = new MyRunnable(obj);
Thread t1=new Thread(r);
Thread t2=new Thread(r);
Thread t3=new Thread(r);
Thread t4=new Thread(r);

t1.start();
t2.start();
t3.start();
t4.start();
try {
    t1.join();
    t2.join();
    t3.join();
    t4.join();
}
catch(Exception e){}
int arr[] = new int[3];
// arr[0] for Type - 1 (Lab Only)
// arr[1] for Type - 2 (Embedded Theory, Lab, Project)
// arr[2] for Type - 3 (Lab and Project)

System.out.println("\nRESULTS : ");
arr=obj.fxn();
if(arr[0]>=arr[1] && arr[0]>=arr[2])
{
    if(arr[0]==arr[1])
    {
        System.out.println("Type - 1 (Lab Only) and Type - 2 (Embedded Theory,
Lab, Project) are tied");
    }
    else if(arr[0]==arr[2])
    {

```

```

        System.out.println("Type - 1 (Lab Only) and Type - 3 (Lab and Project) are
tied");
    }
    else
    {
        System.out.println("Type - 1 (Lab Only) is in majority");
    }
}
else if(arr[1]>=arr[0] && arr[1]>=arr[2])
{
    if(arr[0]==arr[1])
    {
        System.out.println("Type - 1 (Lab Only) and Type - 2 (Embedded Theory,
Lab, Project) are tied");
    }
    else if(arr[1]==arr[2])
    {
        System.out.println("Type - 2 (Embedded Theory, Lab, Project) and Type - 3
(Lab and Project) are tied");
    }
    else
    {
        System.out.println("Type - 2 (Embedded Theory, Lab, Project) is in
majority");
    }
}
else if(arr[2]>=arr[1] && arr[2]>=arr[0])
{
    if(arr[2]==arr[0])
    {
        System.out.println("Type - 1 (Lab Only) and Type - 3 (Lab and Project) are
tied");
    }
    else if(arr[1]==arr[2])
    {
        System.out.println("Type - 2 (Embedded Theory, Lab, Project) and Type - 3
(Lab and Project) are tied");
    }
    else
    {
        System.out.println("Type - 3 (Lab and Project) is in majority");
    }
}
}

```

```
}
```

```
}
```

## OUTPUT :

```
Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac CourseChange.java
Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java CourseChange
-- COURSE TYPE CHANGE FOR CSE1007 - JAVA PROGRAMMING --
Counting started..
```

RESULTS :

```
Total votes of type 1 - Lab Only : 652
Total types of type 2 - Embedded Theory, Lab, Project : 675
Total types of type 3 - Lab and Project : 673

Type - 2 (Embedded Theory, Lab, Project) is in majority
```

```
Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac CourseChange.java
Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java CourseChange
-- COURSE TYPE CHANGE FOR CSE1007 - JAVA PROGRAMMING --
Counting started..
```

RESULTS :

```
Total votes of type 1 - Lab Only : 676
Total types of type 2 - Embedded Theory, Lab, Project : 674
Total types of type 3 - Lab and Project : 650

Type - 1 (Lab Only) is in majority
```

## **15. LAMBDA EXPRESSION TO TRAVERSE ARRAY, LOOPS, INPUT FROM COMMAND LINE ARGUMENTS –**

**Prof. Jaisankar has recently enabled marks on codetantra and is now receiving doubts from some students for re-evaluation. On day i, he receives Qi queries. But due to other academic work, Prof. Jaisankar can answer at most k queries in a single day.**

**He always answers the maximum number of questions that he can on any given day (note however that this cannot be more than k). The remaining questions (if any) will be carried over to the next day. Fortunately, after n days, the queries have stopped and all the doubts are cleared. Prof. Jaisankar would like to know the first day during which he has some free time, i.e. the first day when he answered less than k questions. Use lambda function in this Java code.**

**INPUT : (given as command line arguments, Q1 Q2 Q3 ..... Qn K).**

**OUTPUT :**

**1<sup>st</sup> day when Prof. Jaisankar is free**

### **CODE :**

```
import java.util.*;
import java.lang.*;
import java.io.*;

public class Queries
{
    static int day = 1;
    public static void main (String[] args)
    {
        int i;
        int n = args.length;
        ArrayList<String> queriesList = new ArrayList<String>(Arrays.asList(args));
        queriesList.remove(n-1);

        System.out.println("CAT2 MARKS QUERIES");
        System.out.println("Queries received on each day: ");
    }
}
```

```

// Lambda expression to traverse the array
queriesList.forEach(q -> System.out.println("Day " + (day++) + " : " + q));

System.out.println("Maximum queries that can be answered on a day is " +
args[n-1]);

long[] arr;
long k = Integer.parseInt(args[n-1]);
arr = new long[n];
for (i = 0; i < n - 1; i++) {
    arr[i] = Integer.parseInt(args[i]);
}

long leftover = 0;
for (i = 0; i < n; i++) {
    if(arr[i] > k){
        leftover = arr[i]-k;
        if(i+1 < n) {
            arr[i + 1] += leftover;
        }
    }
    else if (arr[i] < k){
        System.out.println("First day when Prof. Jaisankar was free was day " +
(i+1));
        break;
    }
    if(i==n-1 && leftover >= 0){
        long aa = leftover/k;
        long ans = i+1+aa+1;
        System.out.println("First day when Prof. Jaisankar was free was day " +
(ans));
    }
}
}
}
}

```

## OUTPUT :

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ javac Queries.java
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Queries 10 5 5 3 2 1 5
CAT2 MARKS QUERIES
Queries received on each day:
Day 1 : 10
Day 2 : 5
Day 3 : 5
Day 4 : 3
Day 5 : 2
Day 6 : 1
Maximum queries that can be answered on a day is 5
First day when Prof. Jaisankar was free was day 6
```

```
[Rushabhs-MacBook-Pro:TH-DA-Java rushabh$ java Queries 6 4 2 3 2 1 5
CAT2 MARKS QUERIES
Queries received on each day:
Day 1 : 6
Day 2 : 4
Day 3 : 2
Day 4 : 3
Day 5 : 2
Day 6 : 1
Maximum queries that can be answered on a day is 5
First day when Prof. Jaisankar was free was day 3
```