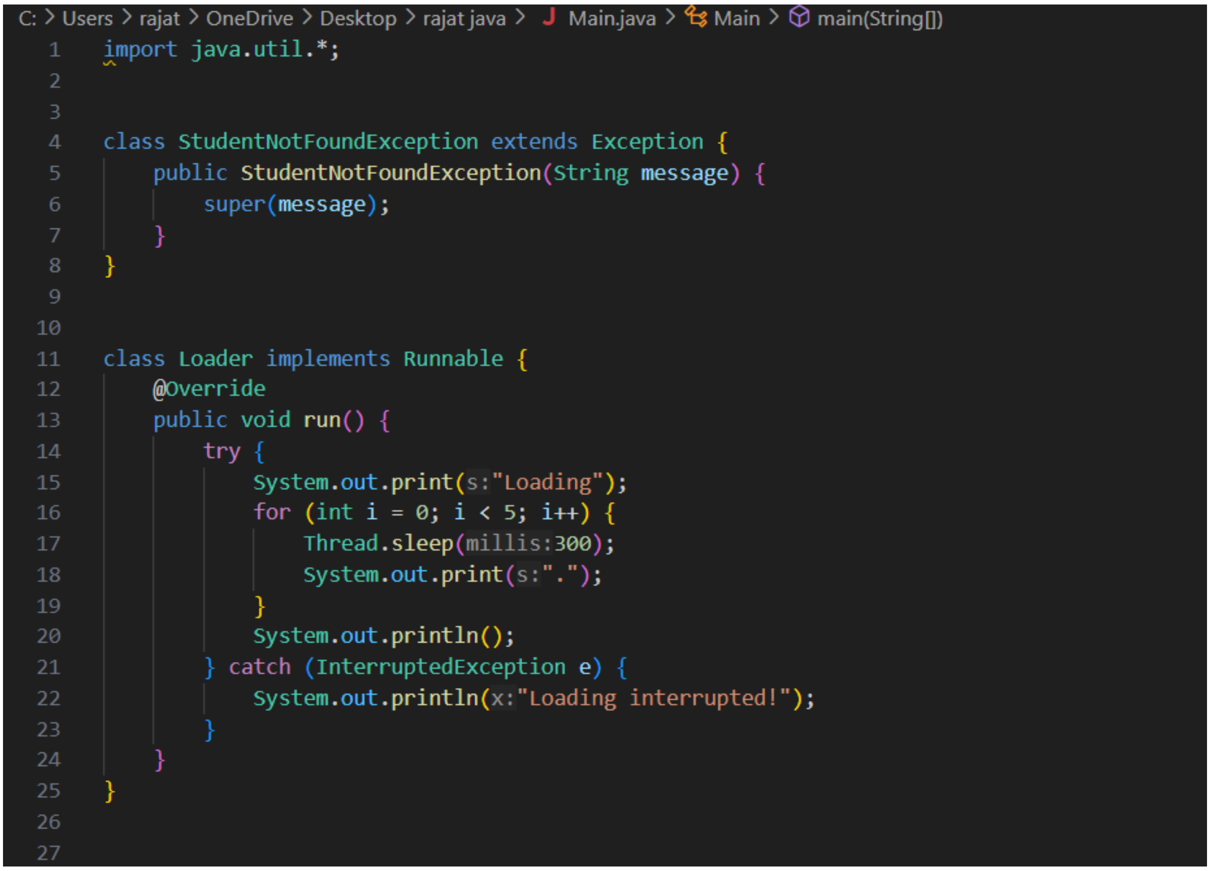
LAB ASSIGNMENT-3

NAME – Rajat Rao

Roll no. – 2401201067

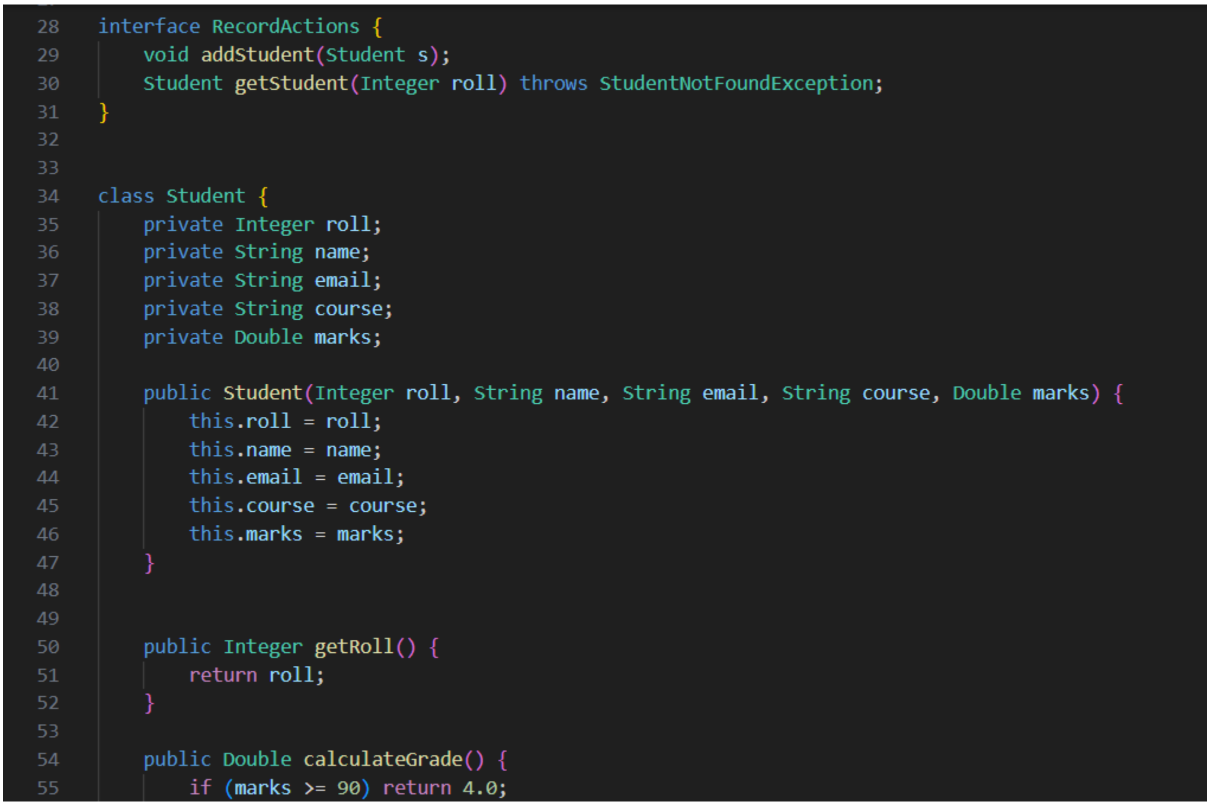
Course – BCA (AI&DS)

INPUT-

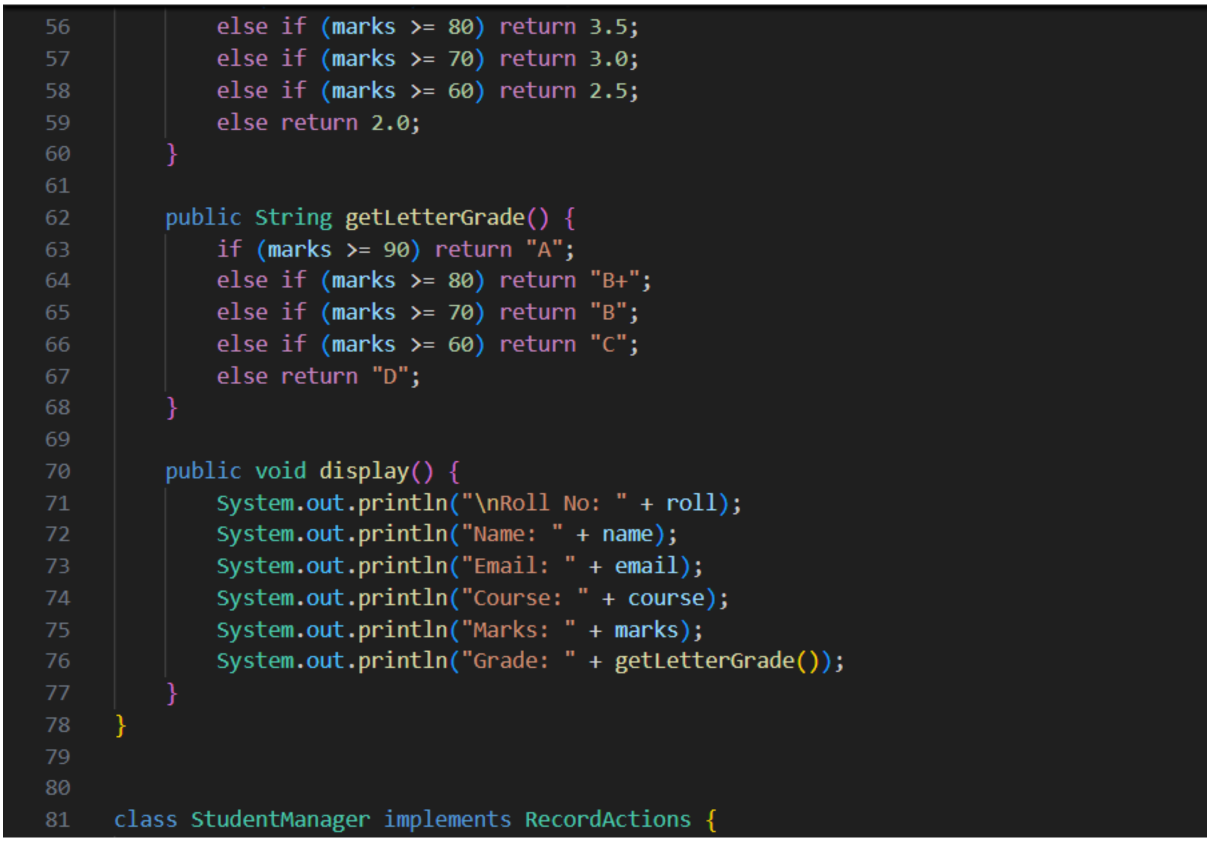


C: > Users > rajat > OneDrive > Desktop > rajat java > Main.java > Main > main(String[])

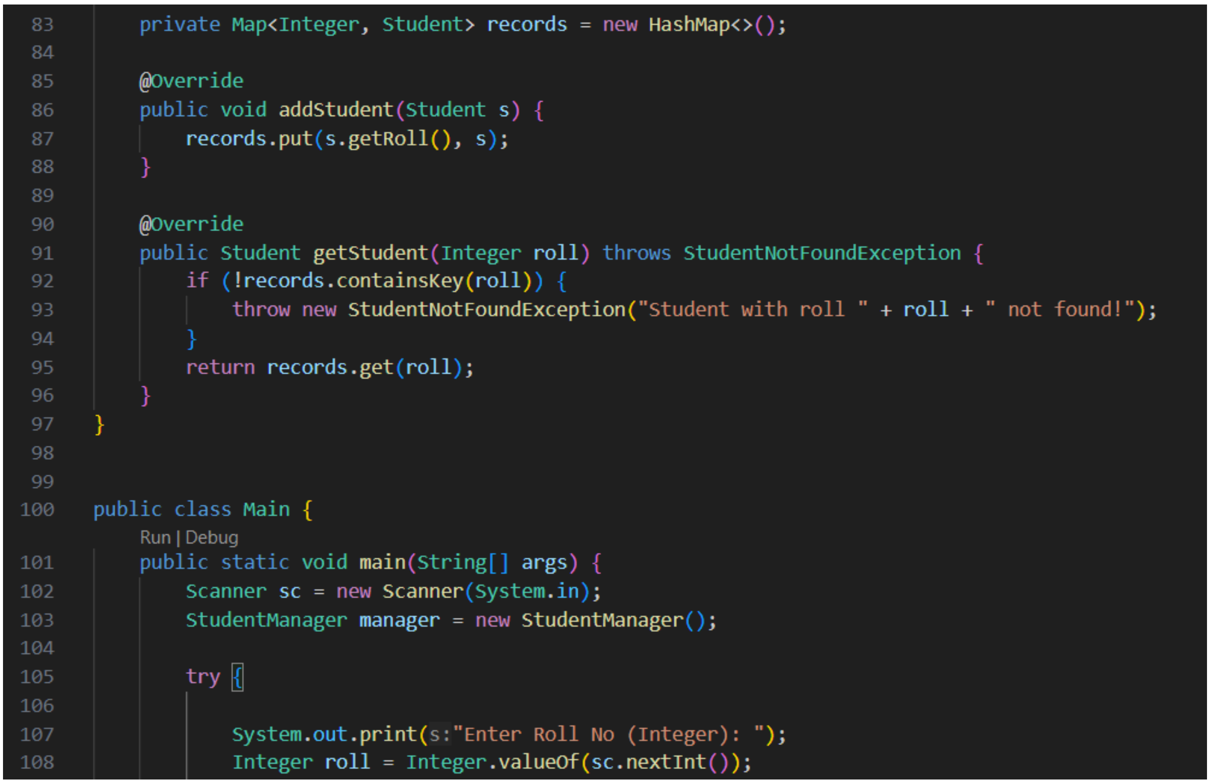
```java
import java.util.*;


class StudentNotFoundException extends Exception {
    public StudentNotFoundException(String message) {
        super(message);
    }
}


class Loader implements Runnable {
    @Override
    public void run() {
        try {
            System.out.print(s:"Loading");
            for (int i = 0; i < 5; i++) {
                Thread.sleep(millis:300);
                System.out.print(s:".");
            }
            System.out.println();
        } catch (InterruptedException e) {
            System.out.println(x:"Loading interrupted!");
        }
    }
}


```



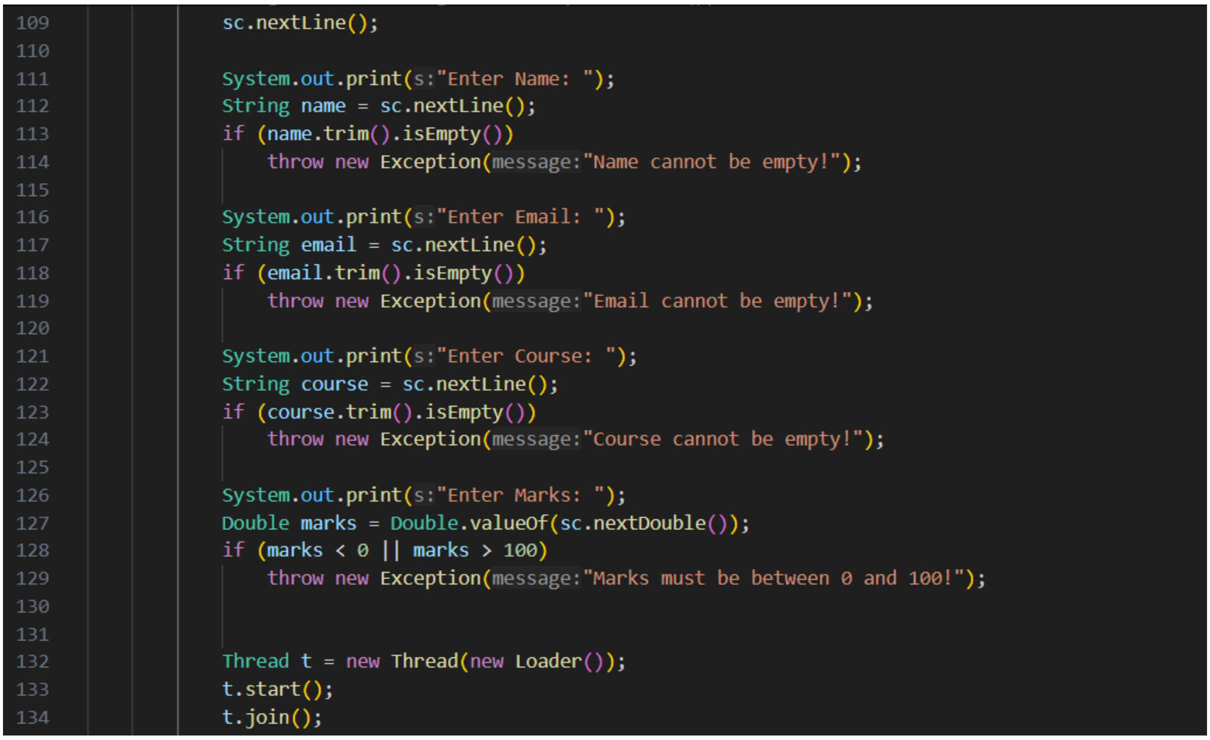
```java
interface RecordActions {
    void addStudent(Student s);
    Student getStudent(Integer roll) throws StudentNotFoundException;
}


class Student {
    private Integer roll;
    private String name;
    private String email;
    private String course;
    private Double marks;

    public Student(Integer roll, String name, String email, String course, Double marks) {
        this.roll = roll;
        this.name = name;
        this.email = email;
        this.course = course;
        this.marks = marks;
    }


    public Integer getRoll() {
        return roll;
    }

    public Double calculateGrade() {
        if (marks >= 90) return 4.0;
```



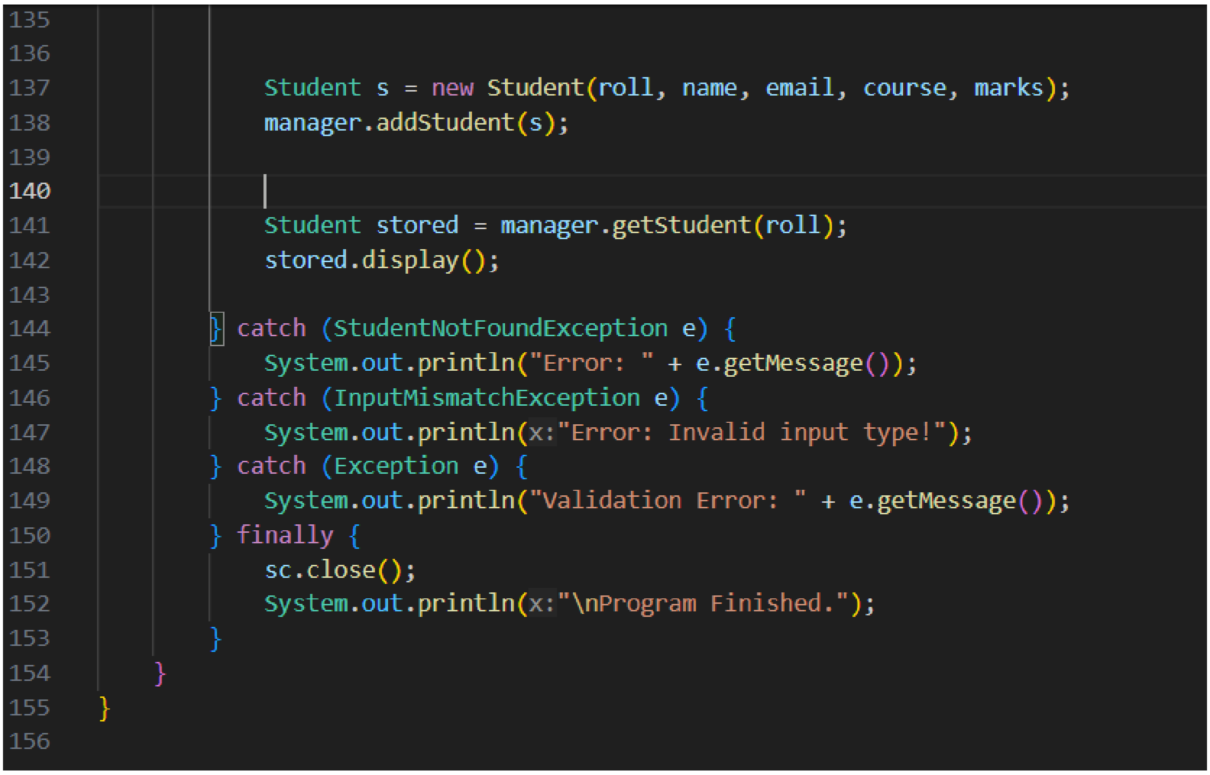
```java
        else if (marks >= 80) return 3.5;
        else if (marks >= 70) return 3.0;
        else if (marks >= 60) return 2.5;
        else return 2.0;
    }

    public String getLetterGrade() {
        if (marks >= 90) return "A";
        else if (marks >= 80) return "B+";
        else if (marks >= 70) return "B";
        else if (marks >= 60) return "C";
        else return "D";
    }

    public void display() {
        System.out.println("\nRoll No: " + roll);
        System.out.println("Name: " + name);
        System.out.println("Email: " + email);
        System.out.println("Course: " + course);
        System.out.println("Marks: " + marks);
        System.out.println("Grade: " + getLetterGrade());
    }
}


class StudentManager implements RecordActions {
```



```
    private Map<Integer, Student> records = new HashMap<>();

    @Override
    public void addStudent(Student s) {
        records.put(s.getRoll(), s);
    }

    @Override
    public Student getStudent(Integer roll) throws StudentNotFoundException {
        if (!records.containsKey(roll)) {
            throw new StudentNotFoundException("Student with roll " + roll + " not found!");
        }
        return records.get(roll);
    }
}


public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StudentManager manager = new StudentManager();

        try {

            System.out.print(s:"Enter Roll No (Integer): ");
            Integer roll = Integer.valueOf(sc.nextInt());
```

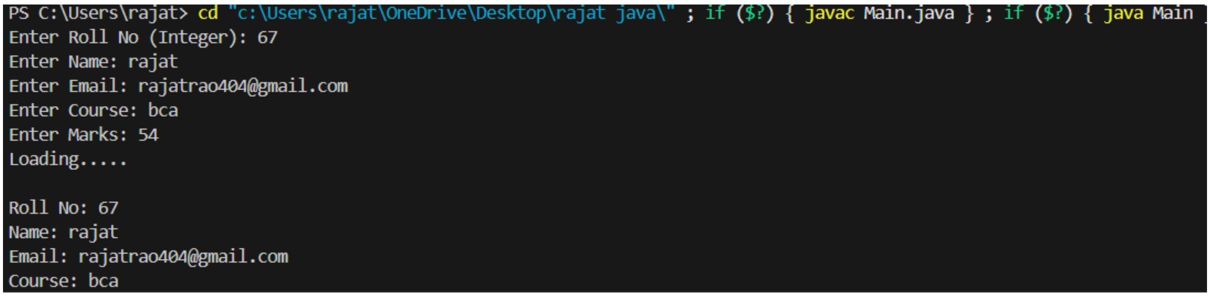


```
            sc.nextLine();

            System.out.print(s:"Enter Name: ");
            String name = sc.nextLine();
            if (name.trim().isEmpty())
                throw new Exception(message:"Name cannot be empty!");

            System.out.print(s:"Enter Email: ");
            String email = sc.nextLine();
            if (email.trim().isEmpty())
                throw new Exception(message:"Email cannot be empty!");

            System.out.print(s:"Enter Course: ");
            String course = sc.nextLine();
            if (course.trim().isEmpty())
                throw new Exception(message:"Course cannot be empty!");

            System.out.print(s:"Enter Marks: ");
            Double marks = Double.valueOf(sc.nextDouble());
            if (marks < 0 || marks > 100)
                throw new Exception(message:"Marks must be between 0 and 100!");


            Thread t = new Thread(new Loader());
            t.start();
            t.join();
```

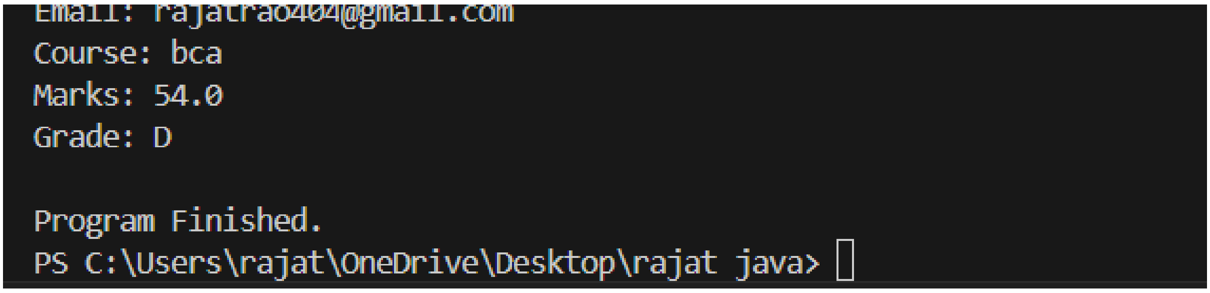


```


            Student s = new Student(roll, name, email, course, marks);
            manager.addStudent(s);


            Student stored = manager.getStudent(roll);
            stored.display();

        } catch (StudentNotFoundException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (InputMismatchException e) {
            System.out.println(x:"Error: Invalid input type!");
        } catch (Exception e) {
            System.out.println("Validation Error: " + e.getMessage());
        } finally {
            sc.close();
            System.out.println(x:"\nProgram Finished.");
        }
    }
}

```

OUTPUT-



```
PS C:\Users\rajat> cd "c:\Users\rajat\OneDrive\Desktop\rajat java\" ; if ($?) { javac Main.java } ; if ($?) { java Main
Enter Roll No (Integer): 67
Enter Name: rajat
Enter Email: rajatrao404@gmail.com
Enter Course: bca
Enter Marks: 54
Loading.....

Roll No: 67
Name: rajat
Email: rajatrao404@gmail.com
Course: bca
```



```
Email: rajatrao404@gmail.com
Course: bca
Marks: 54.0
Grade: D

Program Finished.
PS C:\Users\rajat\OneDrive\Desktop\rajat java> 
```