ASSIGNMENT - 3

Date ....

Name : Divayam Sidhant Yadav
Roll no : 2401730220
Course : Java Programming

1) Custom Exception class

```
package student.result.system;
public class InvalidMarkException extends Exception {
        public InvalidMarkException (String message) {
                super (message);
        }
}
```

2) Student class

```
package student.result.system;
public class student {
        private int rollNumber;
        private string studentName;
        private int [] marks;
        private static final int Min_Marks = 0;
        private static final int Max_Marks = 100;
        private static final int Total_Subject = 3;
        private static final int Passing_Marks = 40;

public student (int rollNumber, string studentName,
                int [] Marks) {
        This. rollNumber = rollNumber;
        This . student Name = studentName;
        This . Marks = Marks. clone ();
}
```

Spiral

Date ....

```
public void validate Marks() Throw Invalid Marks
 Exception {
    if (Marks ==null) {
       throw new Invalid Marks Exception ("Marks
            array can not be null");
    }
    if (marks.length != Total_Subjects) {
        throw new Invalid_Marks Exception ("Exactly" +
              Total_Subject + " Subjects required") .
    }
    for (int i=0 ; i <marks.length ; i++) {
        if (marks [i] < Min_Marks || Marks [i] >
           Max_Marks) {
          throw new Invalid Marks Exception . (
            "Invalid Marks for Subject" + (i+1) + ":" +
           Marks [i] + " Marks must be between " +
           Min_Marks + "and" + Max_Mark );
          }
     }
  }
 public double calculate Average () {
      int total = 0 ·
      for (int Mark : Marks) {
           Total += Marks;
      }
        return (double) Total (Marks.length;
    }
  public boolean in Pass () {
           for (int mark : Marks) {
        if (mark < passing_Marks) {
            return false;
     }
  }
```

Spiral

Date ....

```
public void displayResult (.) {
        Sout ("Roll Number : " + rollNumber);
        Sout (" Student Name; "+ studentName);
        Sout ("Marks:");
        for (int mark : marks) {
              Sout (marks + " ");
        }
          Sout ();
          Sout ("Average : "+ calculate Average ());
          Sout ("Result : " + (is Pass () ? ("Pass" : "fail"));
     }
public int get RollNumber (.) {
            return rollNumber;
  }
  public string getStudentName(){
            return studentName;
  }
   public int [] getMarks () {
          return Marks. clone ();
    }
  }
```

3. Result Manager Class.

```
 package student.result.system;
 import java.util.InputMismatchException;
 import java.util. scanner;
 public class ResultManager {
          private student [] students = new student
                                                 [100];
          private int studentCount = 0;
```

Spiral

Date ....

```
    private Scanner Scanner = new scanner (system.in);

public void mainMenu() {
    int choice;
    do { Sout ("\n===Student Result System===");
        Sout ("1. Add Student");
        Sout ("2. Show Student Details");
        Sout ("3. Exit");
       Sout ("Enter choice: ");
    try { Choice = Scanner.nextInt();
         Switch (choice) {
         case 1: add StudentWith Handling ();
         break;
         case 2: Show Student Details ();
         break;
         case 3: Sout ("Exiting"); break;
         default: Sout ("Invalid choice");
      }

    public static void main (String [] args) {
         Result Manager manager = new Result
              Manager ();
    try {
         manager.Main Menu ();
    }
    finaly {
         manager.Scanner.close();
         Sout ("Program Completed");
      }
  }
}
```

Spiral