Name - Aaneya , 2401730202

Topic _____ Date. _____

Assignment - 3

Course Name → Java Programming

1) Custom Exception class

```
package student. result. system;
public class InvalidMarksException extends Exception
    {
        public InvalidMarksException (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_Subjects = 3;
    private Static final int Passing_Marks = 40;

public Student (int rollNumber, string studentName,
                 int [] Marks) {
        This. rollNumber = rollNumber;
        This. studentName = StudentName;
        This. Marks = Marks. clone();
}
public void validateMarks () Throws InvalidMarks
                                    Exception {
```

Date.

Topic

```
if(Marks == null) {
  throw new InvalidMarksException ("Marks
                     array can not be null");

        }
if(marks.length != Total_Subjects) {
 throw new Invalid_Marks Exception ("Exactly "+
        Total_Subjects + "Subjects required");
  }
 for (int i=0; i < marks.length; i++) {
   if(marks[i] < Min_Marks || Marks[i] >
      Max_Marks) {
   throw new InvalidMarksException (
     "Invalid Marks for subject"+ (i+1) + ":"+
     Marks [i] +". Marks must be between"
     +Min_Marks + "and" + Max_Marks
  );
  }
 }
}
public double calculateAverage () {
   int total = 0;
   for (int Mark : Marks) {
      Total += Marks;
  }
  return (double) Total / Marks.length;
}
public boolean isPass () {
  for (int mark : marks) {
     if (mark < Passing_Marks) {
          return false;
     }
  } return true;
}
```

TELCO

Teacher's Sign.

Topic______________ Date.______

```
public void displayResult() {
    sout ("Roll Number:" + rollNumber);
    Sout (" Student Name :" + studentName);
    sout ("Marks:");
   for (int mark : marks){
        sout (mark+"");
  }
   sout ();
   sout("Average:" + calculateAverage ());
   sout("Result:"+ (isPass() ? "Pass" : "Fail"));
}
public int getRollNumber () {
    return rollNumber;
}
public String getStudentName () {
       return studentName;
}
public int [] getMarks () {
    return Marks . clone ();
}
}
```

3. ResultManager Class

```
package student.result.system ;
import java.util . InputMismatchException ;
import java.util. Scanner;
public class ResultManger {
    private student [] students = new student
                                              [100];
    private int studentCount =0;
    ~~private int Student~~
    private Scanner scanner = new Scanner (system
                                                .in );
```

TELCO Teacher's Sign.............................

Topic ______ Date. ______

```
public void addStudent () throws InvalidMarksException
   { try {
         sout ("Enter Roll number:");
         int rollnumber = Scanner.nextInt();
         Scanner.nextLine ();
         if (findStudent (rollNumber) != null) {
            SOUt ("Error: Student already exists.");
         return;
    }
   SOUt ("Enter Student Name:");
   String name = Scanner.nextline().trim();
   if (name.isEmpty()) throw new IllegalArgum
                              -entException("Name cannot
                              be empty"));
   int [] marks = new int [3];
  for (int i =0; i <3; i++) {
       sout(" Enter marks for subject" +(i+1)
                                        +":");
      marks [i] = Scanner.nextInt();
      marks [i] = Scanner.nextInt ();
      If (marks [i] <0 || marks [i] >100) {
         throw new InvalidMarksException ("Invalid
                               marks:" + marks[i]);
      }
  }
   Student  Student = new Student (rollnumber,
                                name, marks
   Student.validateMarks ();
   if (StudenCount < 100) {
      students [StudentCount ++] = students;
      SOUT ("Student added successfully.");
   } else {
```

TELCO

Teacher's Sign..........................

Topic

```
    sout (" Error : Maximum capacity reached.");
  }
} catch (InputMismatch Exception e) {
      Scanner.nextline();
   throws new InputMismatchException (" Invalid
                     input type.");

   }
}
   private findstudent (int rollnumber) {
      for (i =0; i< studentCount; i++) {
        if (Student [i]. getRollNumber() == roll
                                      Number) {

             return students [i];

        }
     }
   return null;
}
public void showstudentDetails () {
   try {
       sout (" Enter Roll number: ");
       int rollnumber = Scanner.nextInt();
       Student student = findStudent (rollNumber);
       if (Student != null) {
           Student.displayResult ();
       } else {
           sout ("Student not found.");
       }
   } catch (InputMismatchException e) {
       Scanner.nextLine();
       sout ("Error");
   }
}
```

a₁

-2

Topic   Date.

```
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: addStudentWithHandling (); break;
    Case 2: ShowStudentDetails(); Break;
   Case 3: SOUT ("Exiting"); Break;
   default: sout ("Invalid Choice");
  }
 public static void main (string [] args) {
      ResultManager manager = new Result
                                  Manager();
  try {
     manages . MainMenu ();
  } finally {

      manager . scanner . Close ();
      Sout ("Program completed");
  }
 }
}
```

TELCO   Teacher's Sign.......................

Aaneya

Date.

Topic

Assignment 24

Course Name : Java Programming

1) Book Class

```
import java.io.Serializable;
public class Book implement Serializable,
                    comparable <Book> {

   private int BookId ;
   private String title;
   private string author;
   private string category;
   private boolean isIssued;

public Book (int bookId , string title, string author
          , string category) {
       this bookId =  bookId;
       this.title = title
       this author  = author
      this category  = category
       this  isIssued  = false;
   }
public int getBookId () {return BookId;}
public String getAuthor() { return ~~title~~ author
public String gettitle () {return title ;}
public String getcategory () { return category;}
public boolean isIssued  () { return isIssued;}
public void markAsIssued () {this .isIssued
                               = false;}
public void displayBookDetails() {sout ("ID"
   + bookId + " | title" + title + "| Author:"
   + author +" | category" + category + " |
   Status" + (isIssued ?  "Issued" : "Available"));
```

TELCO

Teacher's Sign......................................