< PREV 1 NEXT >

stomer&Address

oding 100 points

CRIPTION

loyee And Address Inheritance(Aggregation)

Create 2 class Employee, Address, EmployeeDemo
dentify the Relationship between Employee & Address class
nd create the code with the below mentioned details of each
lass.
reate getter/setters of Address Class
eate getter/setters of Employee class
eate a class EmployeeDemo and create the below 2
thods in EmployeeDemo Class

a(Employee emp);-->Only Store/accept data

(Employee emp)--->Only for Printing

ut -1

Solution code

Please choose a language and write your code.

ACCEPTED Score: **100 points** (details)

CODE INPUT OUTPUT Java 8 RU

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  // Class name should be "Source",
8  // otherwise solution won't be accepted
9  public class Source {
10     public static void main(String[] args) {
11
12         Scanner s = new Scanner(System.in);
13
14         String a = s.nextLine();
15         String b = s.nextLine();
16         System.out.println("Employee Name :"+b);
```

manipal-adapt.in.capgemini.com is sharing your screen. Stop sharing Hide

Id :"+a);

REDMI NOTE 9 PRO MAX
AI QUAD CAMERA

- Identify the Relationship between Employee & Address class and create the code with the below mentioned details of each class.
- Create getter/setters of Address Class
- Create getter/setters of Employee class
- Create a class EmployeeDemo and create the below 2 methods in EmployeeDemo Class

storeData(Employee emp);-->Only Store/accept data

showData(Employee emp)-->Only for Printing

Sample Input -1

JA1 -----> Employee Id

Jaya ----->Employee Name

Indiranagar -------> Address 1

Rajajinagar -------> Address 2

Bangalore -------> City

123456 -------> Pincode

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

// Class name should be "Source",
// otherwise solution won't be accepted
public class Source {
    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);
        String a = s.nextLine();
        System.out.println("Employee Id :"+a);
        String b = s.nextLine();
        System.out.println("Employee Name :"+b);
        String c = s.nextLine();
        System.out.println("Address 1 :"+c);
        String d = s.nextLine();
        System.out.println("Address 2 :"+d);
        String e = s.nextLine();
        System.out.println("City :"+e);
        String f = s.nextLine();
        System.out.println("Pin :"+f);
    }
}
```


ample Output-1

2 revisions found for this solution.

SHOW RE

“MOBILE SHOP”

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
import java.lang.NullPointerException;
class Mobile{
   // Write your code here..
 HashMap<String, ArrayList<String>> mobiles = new HashMap<>();
 public String addMobile(String company, String model)
{
      if(!mobiles.containsKey(company))
   {
   mobiles.put(company,null);
   }
   if(mobiles.get(company)==null)
   {
    mobiles.put(company, new ArrayList<String>());
   }
       mobiles.get(company).add(model);
 return "model successfully added";
   }
 public ArrayList<String> getModels(String company)
{
 if(mobiles.containsKey(company))
 {
```

```
    return mobiles.get(company);
  }
  return null;
 }
 public String buyMobile(String company, String model)
 {
  try{
  if(mobiles.get(company).contains(model))
     {
         mobiles.get(company).remove(model);
    return "mobile sold successfully";
        }
  }
  catch(NullPointerException e)
  {
   return "item not available";
  }
         return "item not available";
 }
}
public class Source {
 public static void main(String args[] ) throws Exception {
  /* Enter your code here. Read input from STDIN. Print output to STDOUT */
 }
}
```

Catch Me If You Can Program

Catch Me If You Can Program

```
package threadsPractice;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class ExceptionCheck {
 public static void main(String[] args) {
  Scanner a = new Scanner(System.in);
  System.out.println("Enter the input string");
  String inputStr = a.nextLine();
  List<String> outputList = new ArrayList<>();
  Implementation impl = new Implementation();
  outputList = impl.numberCheck(inputStr);
  System.out.println(outputList);

  System.out.println("Enter the file path");
  String inputPath = a.nextLine();
  String fileCheckOutpiut = impl.fileCheck(inputPath);
  System.out.println(fileCheckOutpiut);

 }

}
class Implementation{
```

```
public String fileCheck(String inputPath) {
 try {
   File file = new File(inputPath);
   if(file.exists()) {
    return "File Found";
   }else {
    throw new FileNotFoundException();
   }
      } catch (FileNotFoundException e) {
         e.printStackTrace();
         return e.getMessage();
      }
}
public List<String> numberCheck(String inputStr) {
 String a = null;
 List<String> chars = new ArrayList<>();
 ;
 for (int i = 0; i < inputStr.length(); i++) {
  try {
   if (Character.isDigit(inputStr.charAt(i))) {
    chars.add(String.valueOf(inputStr.charAt(i)));
   } else {
    throw new NumberFormatException();
   }
  } catch (NumberFormatException e) {
   a = "For input string " + "'" + inputStr.charAt(i) + "'";
   chars.add(a);
```

```
  }
 }
 return chars;
}
}
```

Depository – Exception Handling

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
class Repository{
   static String getCountryName(String countryCode) throws InvalidCodeException{
      int num=Integer.parseInt(countryCode);
      if ((num>=70)&&(num<=99))
         return ("India");
      else if(countryCode.equals("011"))
         return ("Dial somewhere else outside of USA");
      else if(num==908)
         return ("US");
      else
         throw new InvalidCodeException("No country with the given code found");
   }
}
class RepositoryImplementation{
   static String getCountry(String countryCode) throws InvalidCodeException{
```

```
        if ((countryCode.length()>3)||(countryCode.length()<2))
            throw new InvalidCodeException("Invalid code detail found");
        else
            return(Repository.getCountryName(countryCode));
        //throw new InvalidCodeException("No country with the given code found");
    }
}
class InvalidCodeException extends Exception{
    public InvalidCodeException(String errorMessage){
        super(errorMessage);
    }
}
public class Main extends RepositoryImplementation
{
    public static void main(String[] args) throws Exception{
        Scanner scan =new Scanner(System.in);
        String s1=scan.nextLine();
        String CC =getCountry(s1);
        System.out.print((getCountry(s1))) ;
    }
}
```

Sherlock Needs Help – StringManipulation Function

```
import java.util.regex.*;
import java.util.ArrayList;
class IdentifyWords {
```

```
    public String getPossibleWords(String s1, String s2) {
        String regex = s1.replace("_", ".");
        String check = regex.toUpperCase();
        String[] possibleStrings = s2.split(":", -1);
        ArrayList<String> result = new ArrayList<String>();
        for (String s : possibleStrings) {
            if (Pattern.matches(check, s.toUpperCase())) {
                result.add(s.toUpperCase());
            }
        }
        if (!result.isEmpty()) {
            String joinedString = String.join(":", result);
            return joinedString;
        } else
            return "Code_Error";
    }
}
import java.util.*;
import java.lang.*;
class GetCode
{
    public int getCodeThroughStrings(String s)
    {
        s = s.replaceAll("\\s","");
        int len = s.length();
        int result=0;
        while (len > 0 || result >= 10)
```

```
        {
            if (len == 0)
            {
                len = result;
                result = 0;
            }
            result += len % 10;
            len /= 10;
        }
        return  result;
    }
}
public class Exp
{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        GetCode gc = new GetCode();
        int len = gc.getCodeThroughStrings(s);
        System.out.println(len);
    }
}
```

AddProduct REMOVEProduct UNIQUE Product

```
import java.lang.*;
import java.util.*;
class Product{
```

```
    List<String> productList = new ArrayList<String>();
    public void addProduct(String pName){
        productList.add(pName);
    }
    public void removeProduct(String pName){
        productList.remove(pName);
    }
    public int uniqueProduct(){
        HashSet<String> hset = new HashSet<String>(productList);
        return hset.size();
    }
}
public class Source {
    public static void main(String[] args) {
        Product p1=new Product();
        p1.addProduct("Pen");
        p1.addProduct("Shirt");
        p1.removeProduct("Shirt");
        p1.addProduct("Pen");
        int count = p1.uniqueProduct();
        System.out.println(count);
    }
}
```

Search Doctor - 2 Errors

Search Doctor - 2 Errors

```
import java.util.ArrayList;
import java.util.Collections;
```

```
import java.util.Comparator;
import java.util.List;
import java.util.Scanner;
import java.util.TreeSet;
class Doctor implements Comparable<Doctor> {
 private String name;
 private String speciality;
 private int experience;
 //CODE HERE
 Doctor(String name, String speciality, int experience) {
    this.name = name;
    this.speciality = speciality;
    this.experience = experience;
  }
  public void setExperience(int experience) {
    this.experience = experience;
  }
  public void setName(String name) {
    this.name = name;
  }
  public void setSpeciality(String speciality) {
    this.speciality = speciality;
  }
  public int getExperience() {
    return experience;
  }
  public String getName() {
```

```
    return name;
  }
  public String getSpeciality() {
    return speciality;
  }
  @Override
  public boolean equals(Object obj) {
    Doctor guest = (Doctor) obj;
    return (guest.name.equals(this.name) && guest.speciality.equals(this.speciality)
        && guest.experience == this.experience);
  }
  @Override
  public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + (this.name == null ? 0 : this.name.hashCode());
    result += prime * result + (this.speciality == null ? 0 : this.speciality.hashCode());
    result += prime * result + this.experience;
    return result;
  }
  @Override
  public int compareTo(Doctor o) {
    return this.name.compareTo(o.name);
  }
 @Override
  public String toString() {
    return "\t" + this.name + "\t" + this.speciality + "\t" + this.experience;
```

```
  }
}
class doctorSorter implements Comparator<Doctor> {
  @Override
  public int compare(Doctor o1, Doctor o2) {
    int value = o1.getSpeciality().compareTo(o2.getSpeciality());
    if (value != 0) {
      return value;
    }
    return o2.getExperience() - o1.getExperience();
  }
}

class DoctorService {
   //DONT MODIFY THIS
 private List<Doctor> doctorsRepository;

 //DONT MODIFY THIS
 public DoctorService(List<Doctor> doctorsRepository) {
  this.doctorsRepository = doctorsRepository;
 }
   //CODE HERE
   List<Doctor> getExperiencedDoctors(int exp) {
    List<Doctor> filteredDoctorsRepository = new ArrayList<Doctor>();
    for (Doctor d : this.doctorsRepository) {
      if (d.getExperience() >= exp) {
        filteredDoctorsRepository.add(d);
```

```
        }
      }
      Collections.sort(filteredDoctorsRepository, new doctorSorter());
      return filteredDoctorsRepository;
    }

    TreeSet<Doctor> getSpecialityDoctor(String spec) {
      TreeSet<Doctor> filteredDoctorsRepository = new TreeSet<Doctor>();
      for (Doctor d : this.doctorsRepository) {
        if (d.getSpeciality().equals(spec)) {
          filteredDoctorsRepository.add(d);
        }
      }
      return filteredDoctorsRepository;
    }

}
public class Source {

  //DON'T MODIFY THIS
  private static String doctorsData;

  //DON'T MODIFY THIS
  static {
   StringBuilder doctors = new StringBuilder();
   doctors.append("Amy-Pediatrics-16;");
   doctors.append("John-Dermatology-10;");
```

```
 doctors.append("David-Dermatology-15;");
 doctors.append("Bob-Pediatrics-6;");
 doctors.append("Cathy-Dermatology-5;");
 doctors.append("Mavis-Pediatrics-11;");
 doctors.append("Robin-Pediatrics-7;");
 doctors.append("Minty-Dermatology-9;");
 doctors.append("Jim-Cardiology-25;");
 doctorsData = doctors.toString();
 }

public static void main(String[] args) {
 //CODE HERE
 Scanner scannner = new Scanner(System.in);
    String listOfDoctors[] = doctorsData.split(";");
    List<Doctor> dArrayList = new ArrayList<Doctor>();
    for (String doct : listOfDoctors) {
      String details[] = doct.split("-");
      dArrayList.add(new Doctor(details[0], details[1], Integer.parseInt(details[2])));
    }
    DoctorService doctorService = new DoctorService(dArrayList);
    List<Doctor> resultArrayList = new ArrayList<Doctor>();
    TreeSet<Doctor> resultSet = new TreeSet<Doctor>();
    int method = scannner.nextInt();
    if (method == 1 || method == 2) {
      if (method == 1) {
        resultArrayList = doctorService.getExperiencedDoctors(scannner.nextInt());
        if (resultArrayList.size() == 0) {
```

```
            System.out.println("No Doctors Found");
            return;
          }
          for (Doctor doctor : resultArrayList) {
            System.out.println(doctor);
          }
          return;
        } else {
          resultSet = doctorService.getSpecialityDoctor(scannner.nextLine());
          if (resultSet.size() == 0) {
            System.out.println("No Doctors Found");
            return;
          }
          for (Doctor doctor : resultSet) {
             System.out.println(doctor);
          }
          return;
        }
      } else {
        System.out.println("Invalid Choice");
      }
  }
}
```

Colour code validator - Exception Warnings in Output

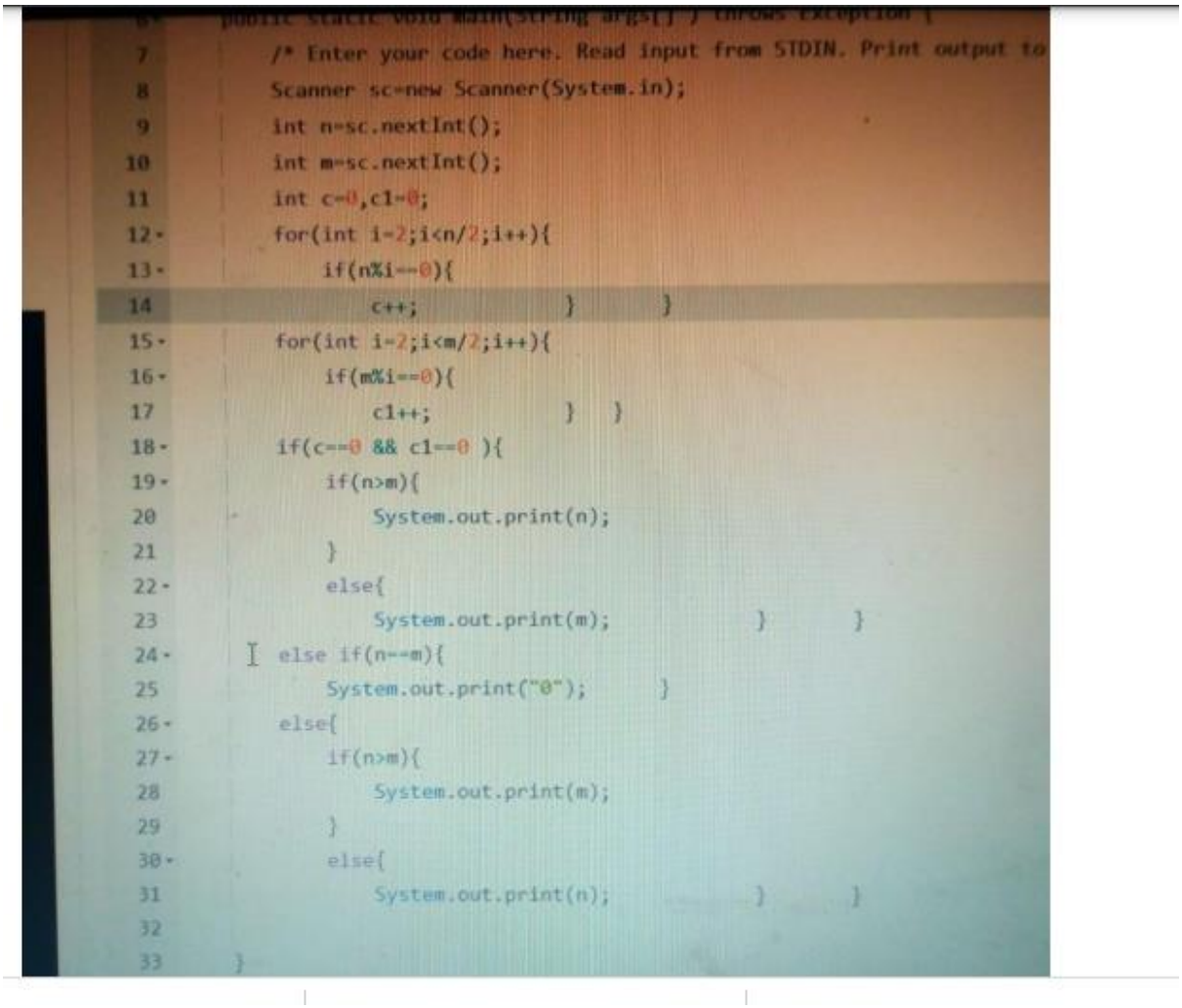
```
import java.util.*;
import java.util.Scanner.*;
class Source {
```

```
public static int validateHexCode(String code) {
   if (code.startsWith("#")) {
      if (code.length() == 7) {
         for (int i = 1; i < code.length(); i++) {
            if ((code.charAt(i) >= 'A' && code.charAt(i) <= 'F')
                  || (code.charAt(i) >= '0' && code.charAt(i) <= '9')) {
               continue;
            } else {
               return -1;
            }
         }
         return 1;
      }
   }
   return -1;
}
public static int validateDecimalCode(String code) {
   if (code.startsWith("rgb(") && code.endsWith(")")) {
      code = code.substring(4, code.length() - 1);
      String ranges[] = code.split(",");
      for (String str : ranges) {
         try {
            int number = Integer.parseInt(str);
            if (number >= 0 && number <= 255) {
               continue;
            } else {
               return -1;
```

```
            }
        } catch (Exception e) {
            return -1;
        }
    }
    return 1;
  }
  return -1;
}
public static void main(String[] args) {
  Scanner scanner = new Scanner(System.in);
  int ch = scanner.nextInt();
  scanner.nextLine();
  if (ch == 1) {
    int r = validateHexCode(scanner.nextLine());
    if (r == 1) {
      System.out.println("Valid Code");
    } else {
      System.out.println("Invalid Code");
    }
  } else if (ch == 2) {
    int r = validateDecimalCode(scanner.nextLine());
    if (r == 1) {
      System.out.println("Valid Code");
    } else {
      System.out.println("Invalid Code");
    }
```

```
        } else {
            System.out.println("Invalid choice");
        }
    }
}
```

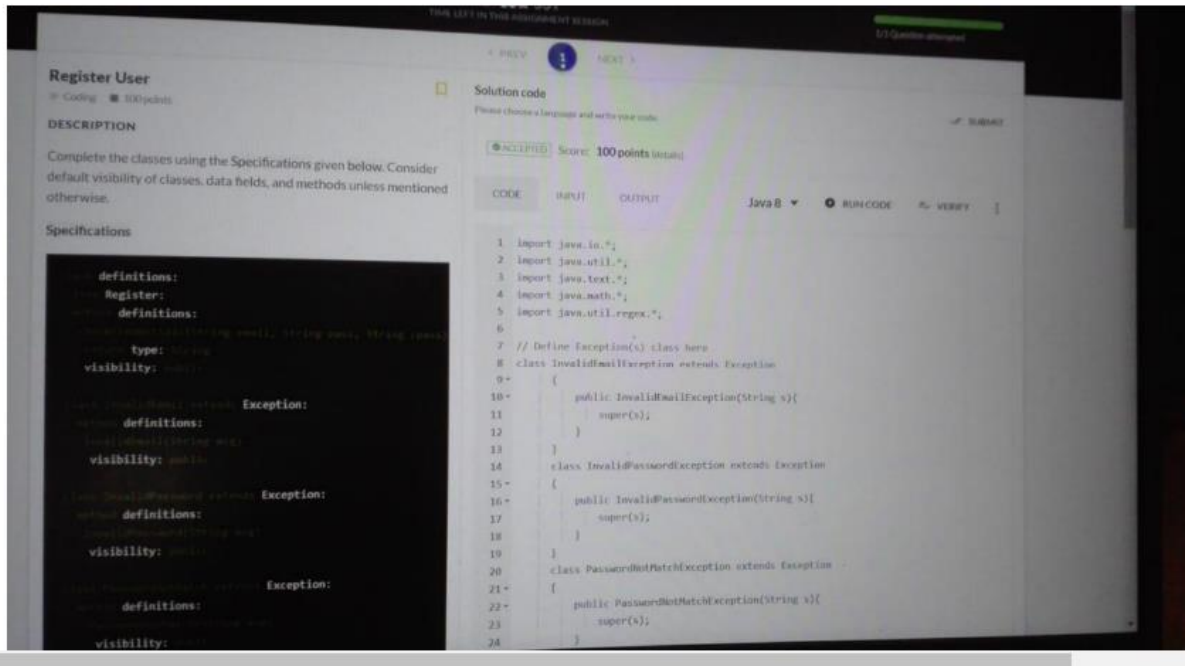
PRIME LARGEST

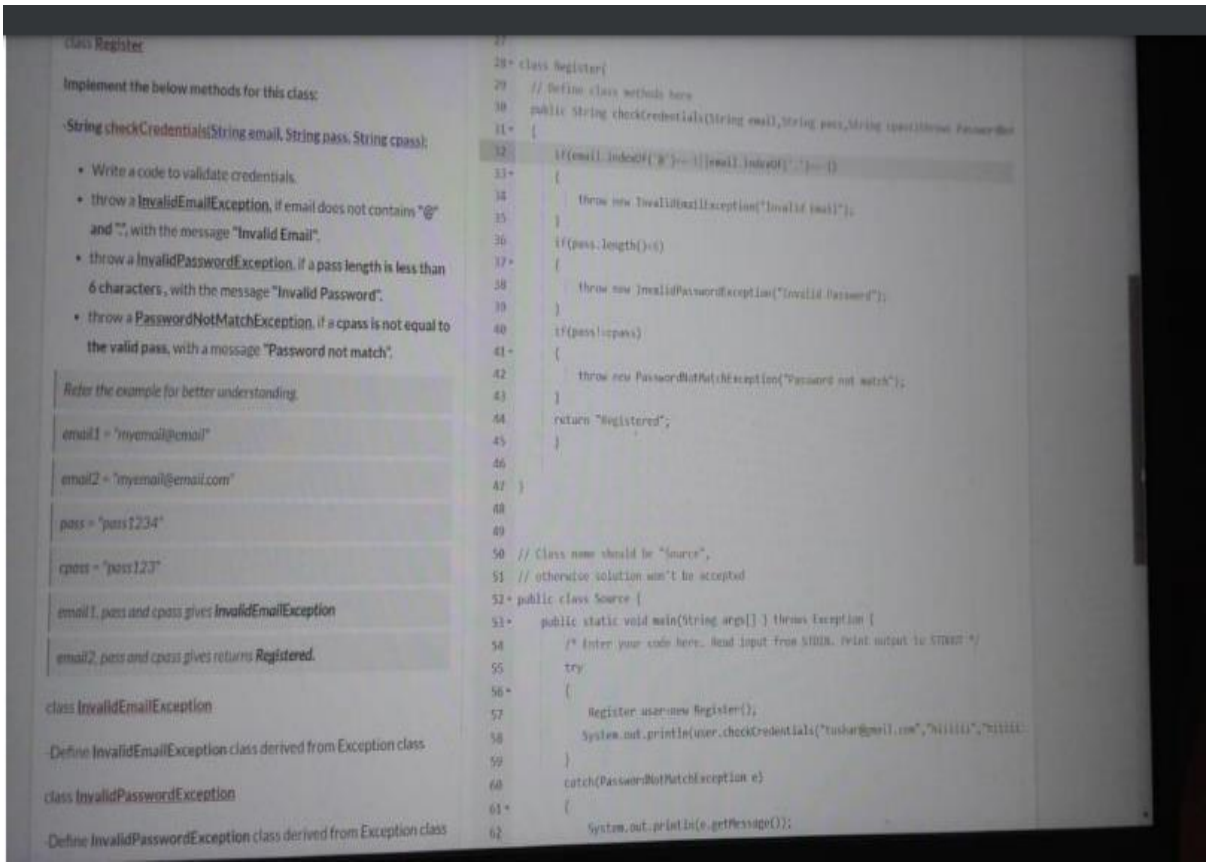


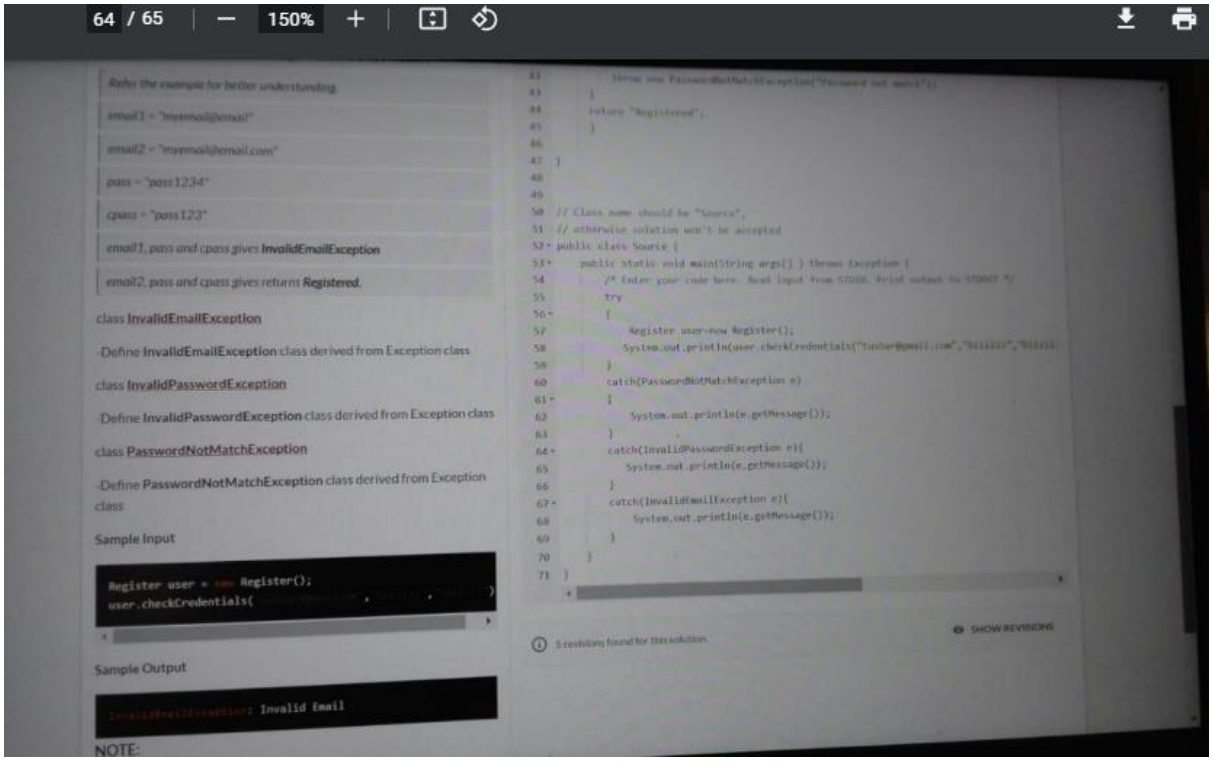
```
7       /* Enter your code here. Read input from STDIN. Print output to
8       Scanner sc=new Scanner(System.in);
9       int n=sc.nextInt();
10      int m=sc.nextInt();
11      int c=0,c1=0;
12      for(int i=2;i<n/2;i++){
13          if(n%i==0){
14              c++;            }       }
15      for(int i=2;i<m/2;i++){
16          if(m%i==0){
17              c1++;           }   }
18      if(c==0 && c1==0 ){
19          if(n>m){
20              System.out.print(n);
21          }
22          else{
23              System.out.print(m);            }       }
24      else if(n==m){
25          System.out.print("0");       }
26      else{
27          if(n>m){
28              System.out.print(m);
29          }
30          else{
31              System.out.print(n);            }       }
32
33  }
```

REGISTER USER

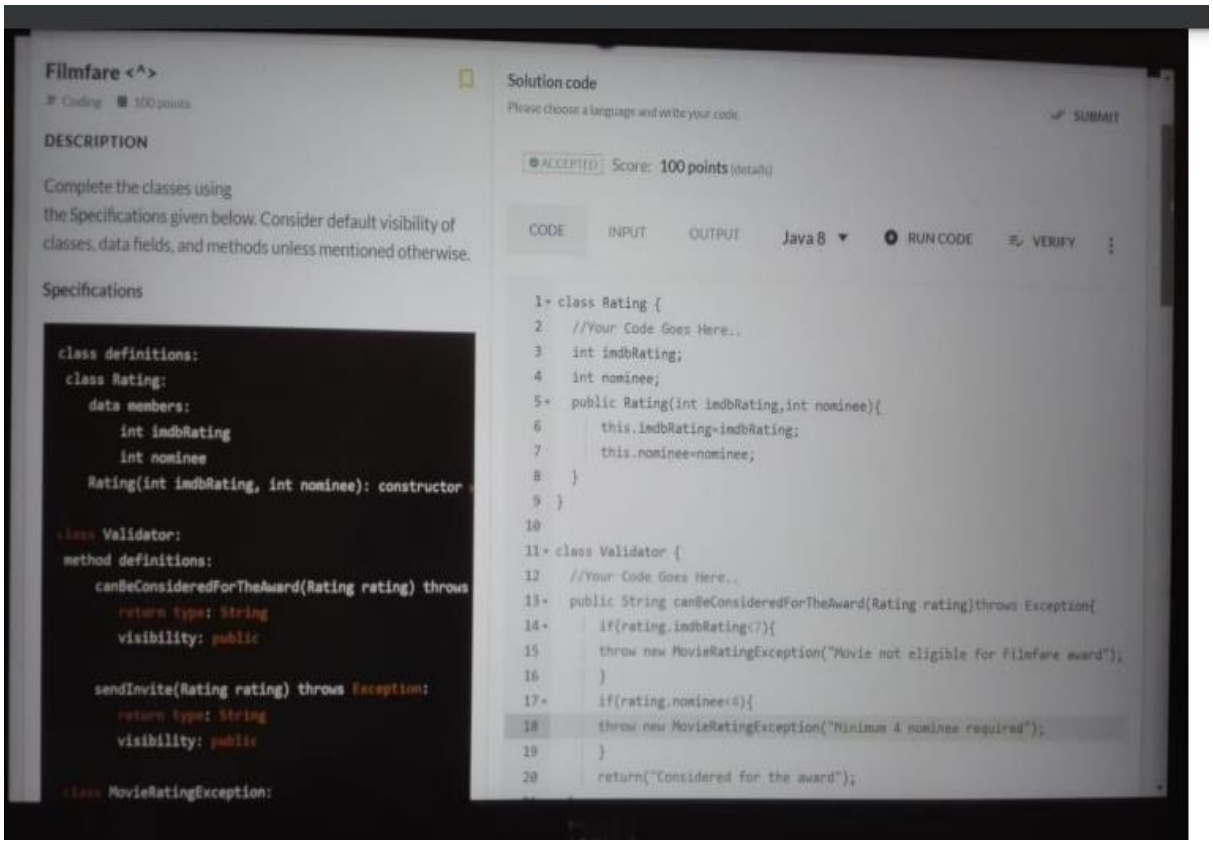
Register User

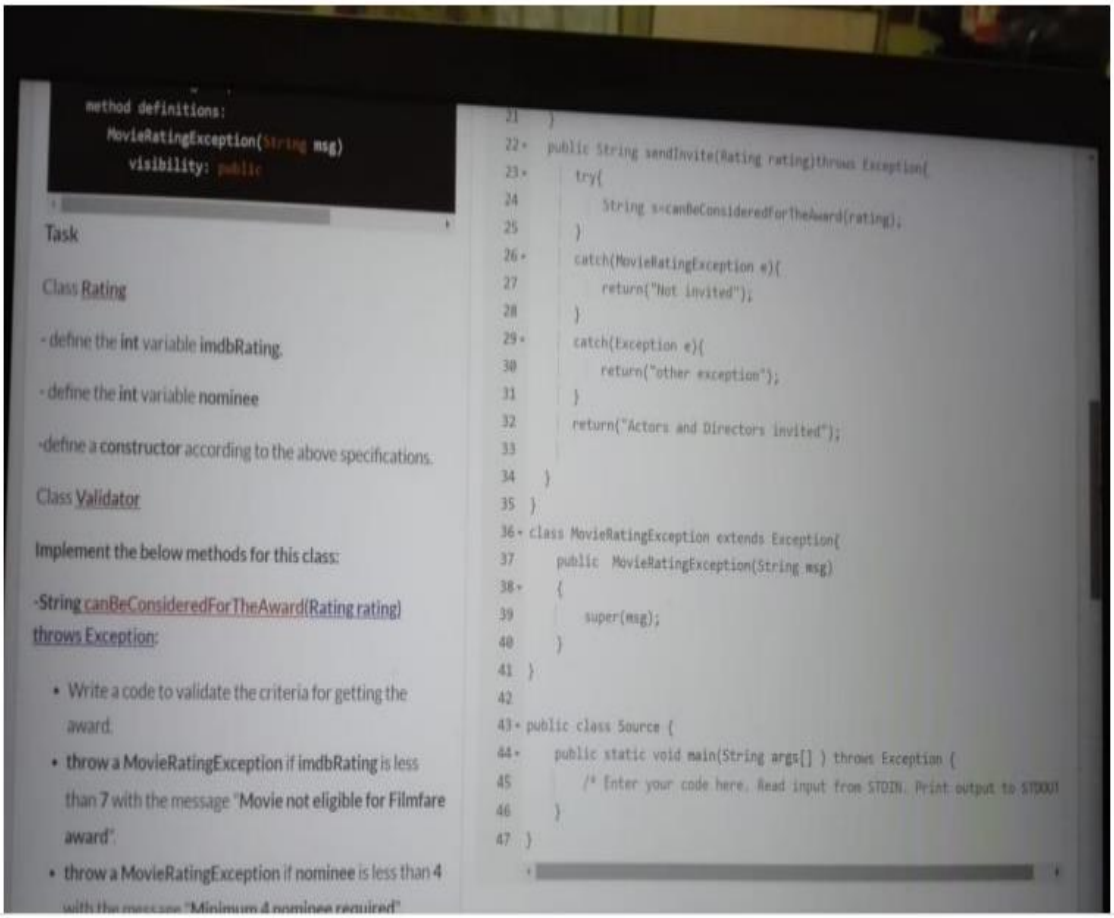





64 / 65
150%
Refer the example for better understanding
email1 = "invemail@email"
email2 = "invemail@email.com"
pass = "pass1234"
cpass = "pass123"
email1, pass and cpass gives InvalidEmailException
email2, pass and cpass gives returns Registered.
class InvalidEmailException
-Define InvalidEmailException class derived from Exception class
class InvalidPasswordException
-Define InvalidPasswordException class derived from Exception class
class PasswordNotMatchException
-Define PasswordNotMatchException class derived from Exception class
Sample Input
Register user = new Register();
user.checkCredentials(
Sample Output
: Invalid Email
NOTE:
5 revisions found for this solution
SHOW REVISIONS

FILMFARE


Filmfare <^>
Coding
100 points
DESCRIPTION
Complete the classes using
the Specifications given below. Consider default visibility of
classes, data fields, and methods unless mentioned otherwise.
Specifications
class definitions:
class Rating:
data members:
int imdbRating
int nominee
Rating(int imdbRating, int nominee): constructor
class Validator:
method definitions:
canBeConsideredForTheAward(Rating rating) throws
return type: String
visibility: public
sendInvite(Rating rating) throws Exception:
return type: String
visibility: public
class MovieRatingException:
Solution code
Please choose a language and write your code
SUBMIT
ACCEPTED
Score: 100 points (details)
CODE
INPUT
OUTPUT
Java 8
RUN CODE
VERIFY
1 class Rating {
2 //Your Code Goes Here..
3 int imdbRating;
4 int nominee;
5 public Rating(int imdbRating,int nominee){
6 this.imdbRating=imdbRating;
7 this.nominee=nominee;
8 }
9 }
10
11 class Validator {
12 //Your Code Goes Here..
13 public String canBeConsideredForTheAward(Rating rating)throws Exception{
14 if(rating.imdbRating<7){
15 throw new MovieRatingException("Movie not eligible for Filmfare award");
16 }
17 if(rating.nominee<4){
18 throw new MovieRatingException("Minimum 4 nominee required");
19 }
20 return("Considered for the award");



method definitions:
MovieRatingException(String msg)
visibility: public

Task

Class Rating

- define the int variable imdbRating.

- define the int variable nominee

-define a constructor according to the above specifications.

Class Validator

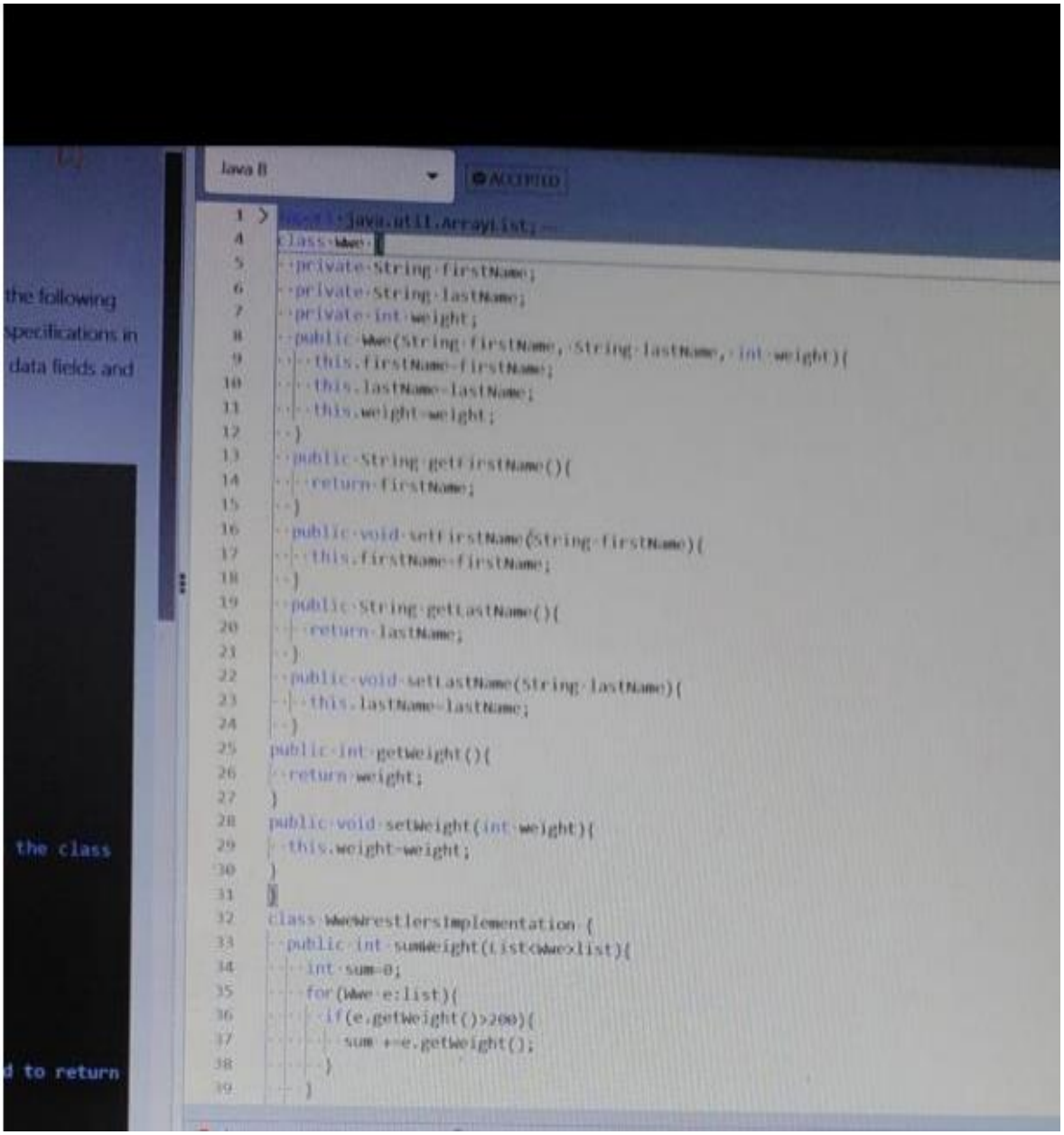
Implement the below methods for this class:

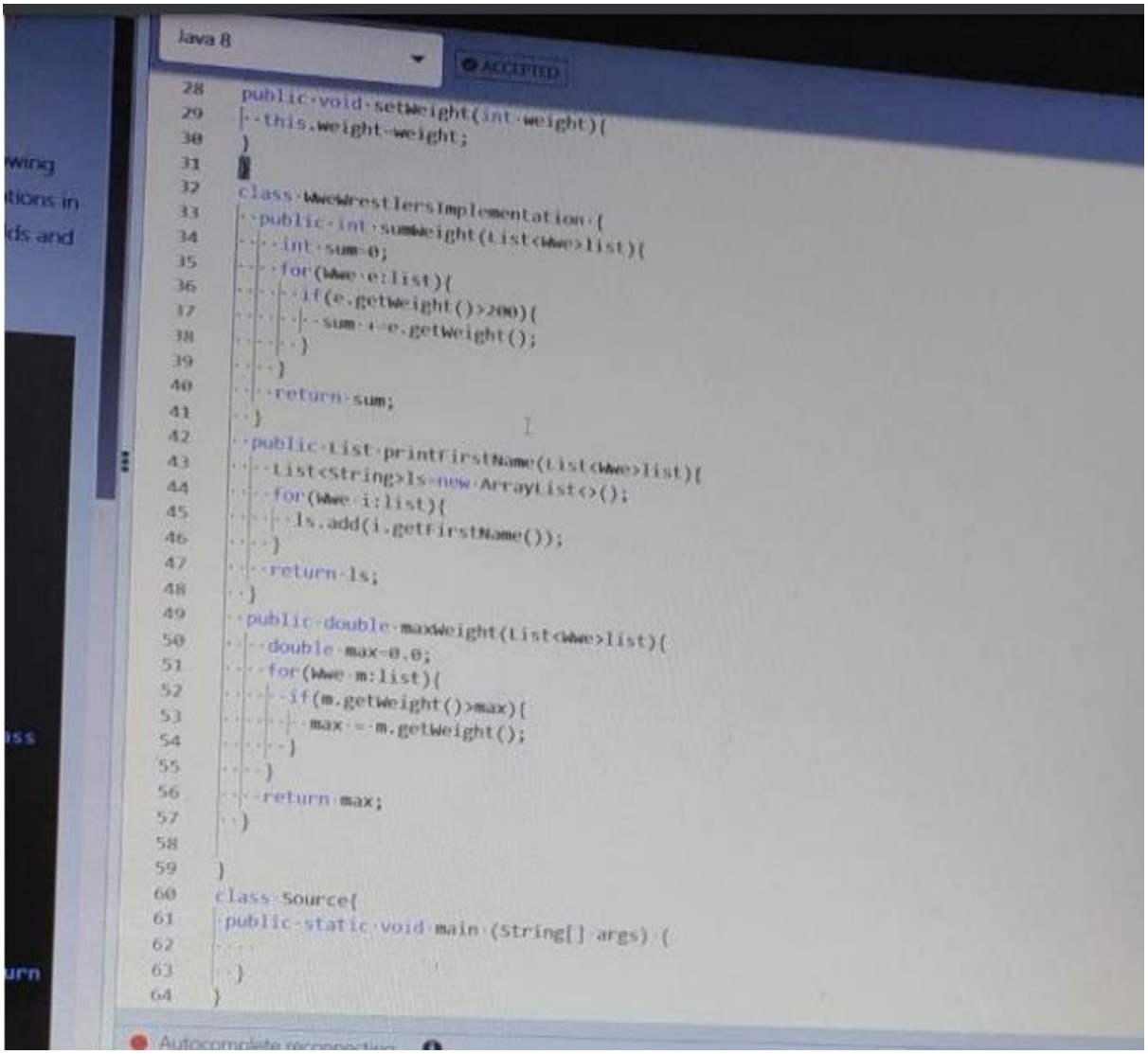
-String canBeConsideredForTheAward(Rating rating) throws Exception:

- Write a code to validate the criteria for getting the award.
- throw a MovieRatingException if imdbRating is less than 7 with the message "Movie not eligible for Filmfare award".
- throw a MovieRatingException if nominee is less than 4 with the message "Minimum 4 nominee required".

```
21    }
22    public String sendInvite(Rating rating)throws Exception{
23        try{
24            String s=canBeConsideredForTheAward(rating);
25        }
26        catch(MovieRatingException e){
27            return("Not invited");
28        }
29        catch(Exception e){
30            return("other exception");
31        }
32        return("Actors and Directors invited");
33
34    }
35 }
36 class MovieRatingException extends Exception{
37     public  MovieRatingException(String msg)
38     {
39         super(msg);
40     }
41 }
42
43 public class Source {
44     public static void main(String args[] ) throws Exception {
45         /* Enter your code here. Read input from STDIN. Print output to STDOUT
46     }
47 }
```

WRESTLEIMPLEMENTATION


the following
specifications in
data fields and
the class
d to return
Java 8
ACCEPTED
import java.util.ArrayList;
class Wwe{
private String firstName;
private String lastName;
private int weight;
public Wwe(String firstName, String lastName, int weight){
this.firstName=firstName;
this.lastName=lastName;
this.weight=weight;
}
public String getFirstName(){
return firstName;
}
public void setFirstName(String firstName){
this.firstName=firstName;
}
public String getLastName(){
return lastName;
}
public void setLastName(String lastName){
this.lastName=lastName;
}
public int getWeight(){
return weight;
}
public void setWeight(int weight){
this.weight=weight;
}
}
class WweWrestlersImplementation {
public int sumWeight(List<Wwe>list){
int sum=0;
for(Wwe e:list){
if(e.getWeight()>200){
sum +=e.getWeight();
}
}


Java 8
ACCEPTED
public void setWeight(int weight){
this.weight=weight;
}
class WweWrestlersImplementation {
public int sumWeight(List<Wwe>list){
int sum=0;
for(Wwe e:list){
if(e.getWeight()>200){
sum+=e.getWeight();
}
}
return sum;
}
public List printFirstName(List<Wwe>list){
List<String>ls=new ArrayList<>();
for(Wwe i:list){
ls.add(i.getFirstName());
}
return ls;
}
public double maxWeight(List<Wwe>list){
double max=0.0;
for(Wwe m:list){
if(m.getWeight()>max){
max=m.getWeight();
}
}
return max;
}
}
class Source{
public static void main (String[] args) {
}
}

5_6330158805002748909.txt

```
import java.util.ArrayList;

public class modifyinglist {

	public ArrayList<String>createList(int num,String str)
	{
		ArrayList<String>arraylist=new ArrayList<String>();

		for(int i=0;i<num;i++)
		{
			arraylist.add(str);
		}
		return arraylist;
	}
	public ArrayList<String>changeListoccurences(ArrayList<String>liststr,String m,String n)
	{
	for(int i=0;i<liststr.size();i++){
		if(liststr.get(i).equals(m)) {
			liststr.set(i, n);
		}
	}
		return liststr;

	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub

	}

}
```

3h 51m left

Help

All

1

2

3

4

Your task here is to implement a **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

Specifications:

```
class definitions:
 class Implementation:
    method definitons:
      changeOccurrence(ArrayList<String> list,String a
          return type: ArrayList<String>
          visibility: public

     getIndex(ArrayList<String> list):
         return type: String
         visibility: public

    addAfter(ArrayList<String> list, String a, Strin
        return type: ArrayList<String>
        visibility: public
```

Task:

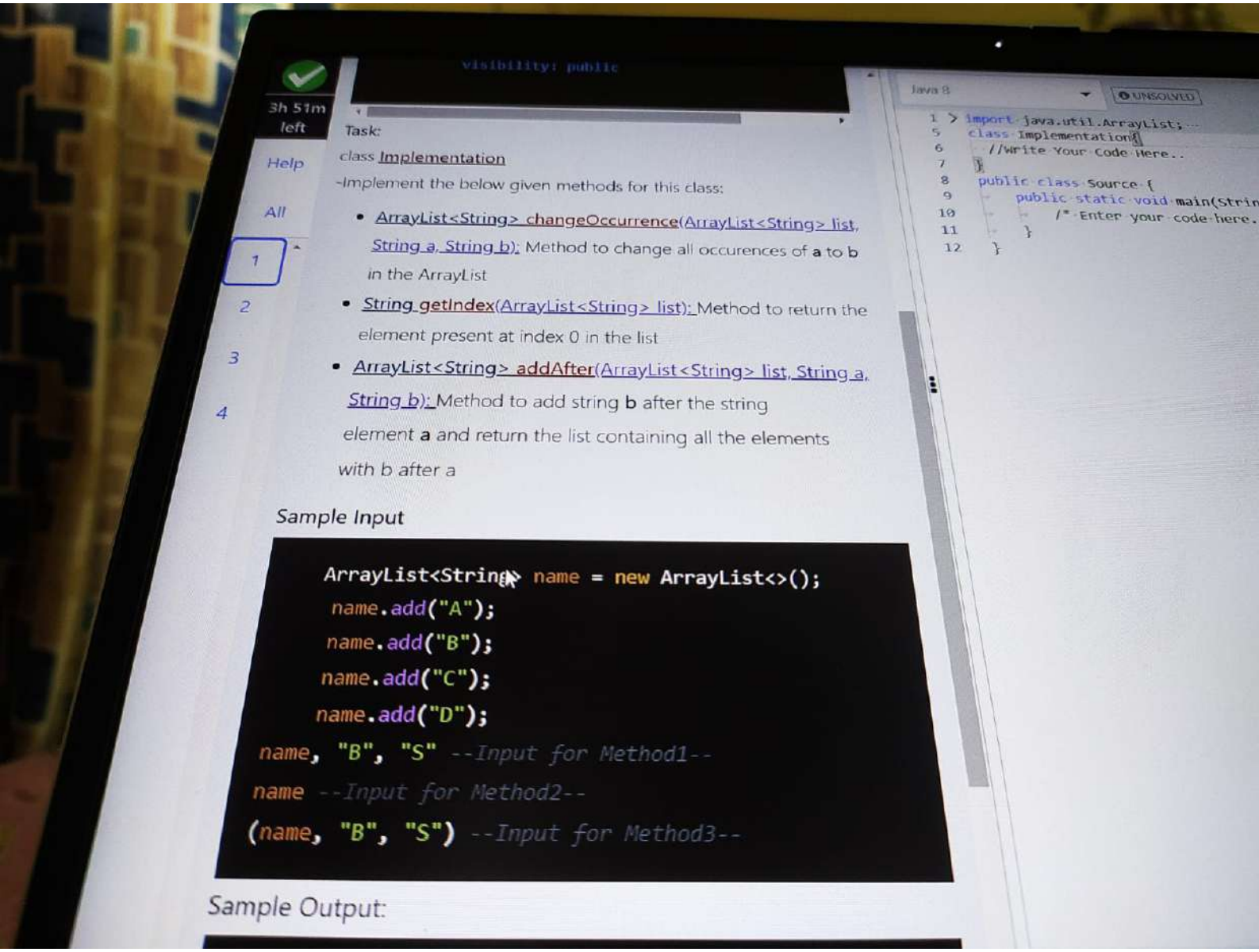
class Implementation

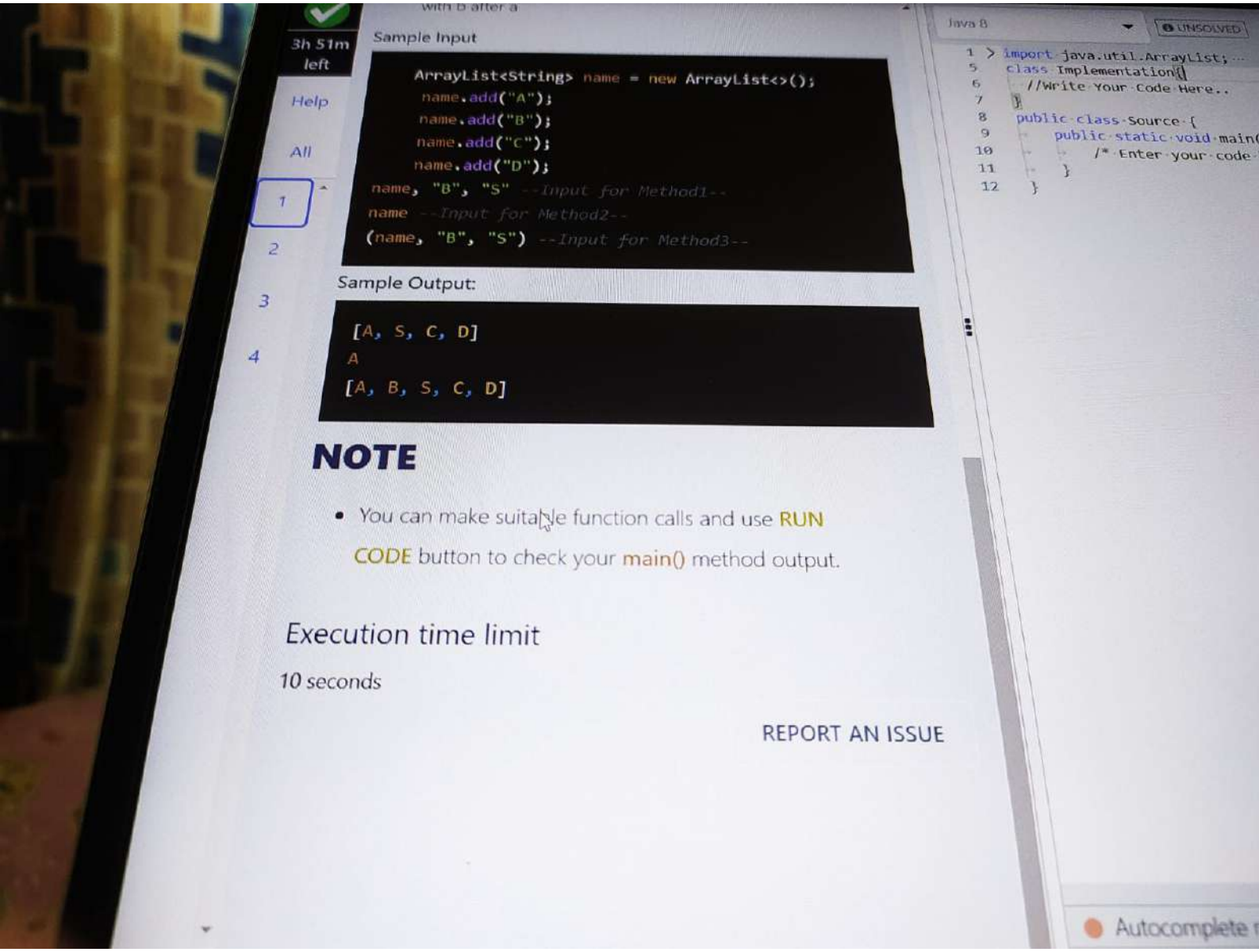
-Implement the below given methods for this class:

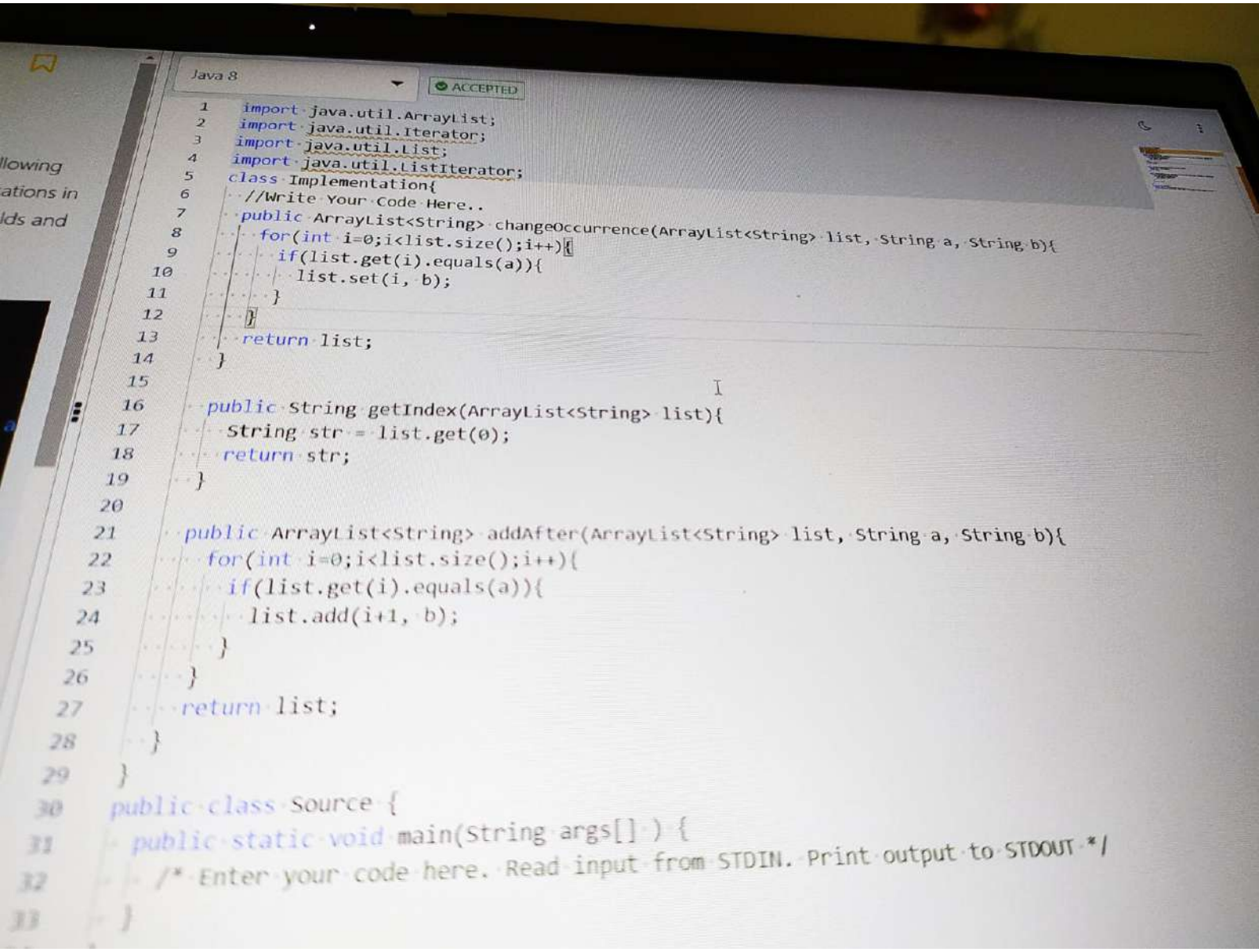
- ArrayList<String> changeOccurrence(ArrayList<String> list, String a, String b): Method to change all occurences of a to b in the ArrayList

UNSOLVED

```
1  import java.util.ArrayList;...
5  class Implementation{
6    //Write Your Code Here..
7  }
8  public class Source {
9      public static void main(Str
10         /* Enter your code here
11     }
12 }
```


visibility: public
3h 51m left
Help
All
1
2
3
4
Task:
class Implementation
-Implement the below given methods for this class:
ArrayList<String> changeOccurrence(ArrayList<String> list, String a, String b): Method to change all occurences of a to b in the ArrayList
String getIndex(ArrayList<String> list): Method to return the element present at index 0 in the list
ArrayList<String> addAfter(ArrayList<String> list, String a, String b): Method to add string b after the string element a and return the list containing all the elements with b after a
Sample Input
ArrayList<String> name = new ArrayList<>();
name.add("A");
name.add("B");
name.add("C");
name.add("D");
name, "B", "S" --Input for Method1--
name --Input for Method2--
(name, "B", "S") --Input for Method3--
Sample Output:
Java 8
UNSOLVED
1 import java.util.ArrayList;...
5 class Implementation{
6 //Write Your Code Here..
7 }
8 public class Source {
9 public static void main(Strin
10 /* Enter your code here.
11 }
12 }

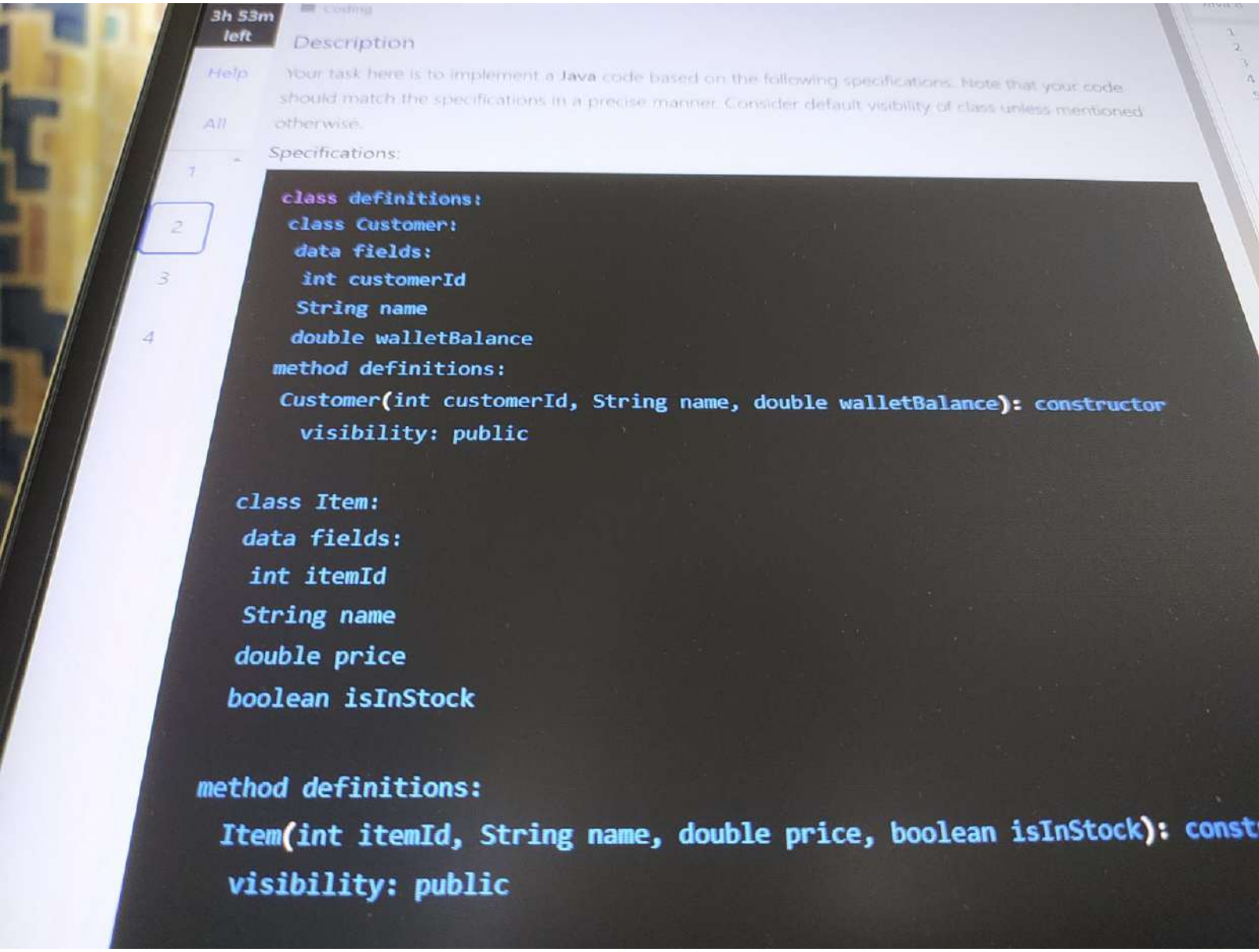

with B after a
3h 51m left
Help
All
1
2
3
4
Sample Input
ArrayList<String> name = new ArrayList<>();
name.add("A");
name.add("B");
name.add("C");
name.add("D");
name, "B", "S" --Input for Method1--
name --Input for Method2--
(name, "B", "S") --Input for Method3--
Sample Output:
[A, S, C, D]
A
[A, B, S, C, D]
NOTE
You can make suitable function calls and use RUN CODE button to check your main() method output.
Execution time limit
10 seconds
REPORT AN ISSUE
Java 8
UNSOLVED
1 import java.util.ArrayList;...
5 class Implementation{
6 //Write Your Code Here..
7 }
8 public class Source {
9 public static void main(
10 /* Enter your code
11 }
12 }
Autocomplete



Java 8

ACCEPTED

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.ListIterator;
class Implementation{
    //Write Your Code Here..
    public ArrayList<String> changeOccurrence(ArrayList<String> list, String a, String b){
        for(int i=0;i<list.size();i++){
            if(list.get(i).equals(a)){
                list.set(i, b);
            }
        }
        return list;
    }

    public String getIndex(ArrayList<String> list){
        String str = list.get(0);
        return str;
    }

    public ArrayList<String> addAfter(ArrayList<String> list, String a, String b){
        for(int i=0;i<list.size();i++){
            if(list.get(i).equals(a)){
                list.add(i+1, b);
            }
        }
        return list;
    }
}
public class Source {
    public static void main(String args[] ) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    }
```


3h 53m left
Description
Help
All
Your task here is to implement a Java code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of class unless mentioned otherwise.
Specifications:
class definitions:
class Customer:
data fields:
int customerId
String name
double walletBalance
method definitions:
Customer(int customerId, String name, double walletBalance): constructor
visibility: public
class Item:
data fields:
int itemId
String name
double price
boolean isInStock
method definitions:
Item(int itemId, String name, double price, boolean isInStock): const
visibility: public

3h 52m left

Help

All

1

2

3

4

```
    Item(int itemId, String name, double price, boolean isInStock): constructor
     visibility: public

  class ShoppingWebsite:
   method definition:
      purchaseItem(Item i, Customer c) throws StockBalanceException:
         return type: String
         visibility: public
class StockBalanceException extends Exception:
 method definition:
    StockBalanceException(String message):
       visibility: public
```

Task:

-Implement class Customer according to the above specifications

-Implement class Item according to the above specifications

-Class ShoppingWebsite

Implement the below given method for this class:

-String purchaseItem(Item i, Customer c) throws StockBalanceException:

- Throw StockBalanceException when the item is out of stock with the message "item is out of stock".
- Throw StockBalanceException when customer wallet balance is not sufficient(Item price is greater than the wallet balance) with the message "customer wallet balance is not sufficient".
- If no exception found then return "Order Successful".

-Class StockBalanceException

- define custom exception class StockBalanceException by extending the Exception class.
- define a parameterized constructor with a String argument to pass the message to the super

3h 52m left

Help

All

- If no exception found then return "Order Successful".

-Class StockBalanceException

- define custom exception class **StockBalanceException** by extending the **Exception** class.
- define a parameterized constructor with a String argument to pass the message to the super class.

Sample Testcase

Input

```
Customer cusDet = new Customer(927392, "Steve" ,5000.0);
Item itemDet = new Item(27392, "T-Shirt", 800, true);
---------------------------------------------------------------
purchaseItem(itemDet, cusDet);
```

output

```
"Order Successful"
```

NOTE

- You can make suitable function calls and use the RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN IS

the following
specifications in
less mentioned

```
3   import java.text.*;
4   import java.math.*;
5   import java.util.regex.*;
6   class Customer {
7       //Write Your Code Here..
8       int customerId;
9       String name;
10      double walletBalance;
11
12      public Customer(int customerId, String name, double walletBalance){
13          this.customerId=customerId;
14          this.name=name;
15          this.walletBalance=walletBalance;
16      }
17  }
18
19  class Item {
20      int itemId;
21      String name;
22      double price;
23      boolean isInStock;
24      //Write Your Code Here..
25      public Item(int itemId, String name, double price, boolean isInStock){
26          this.itemId = itemId;
27          this.name=name;
28          this.price=price;
29          this.isInStock=isInStock;
30      }
31  }
32
33  class ShoppingWebsite {
34      //Write Your Code Here..
35      public String purchaseItem(Item i, Customer c) throws StockBalanceException{
36          if(!i.isInStock){
37              throw new StockBalanceException("item is out of stock");
```

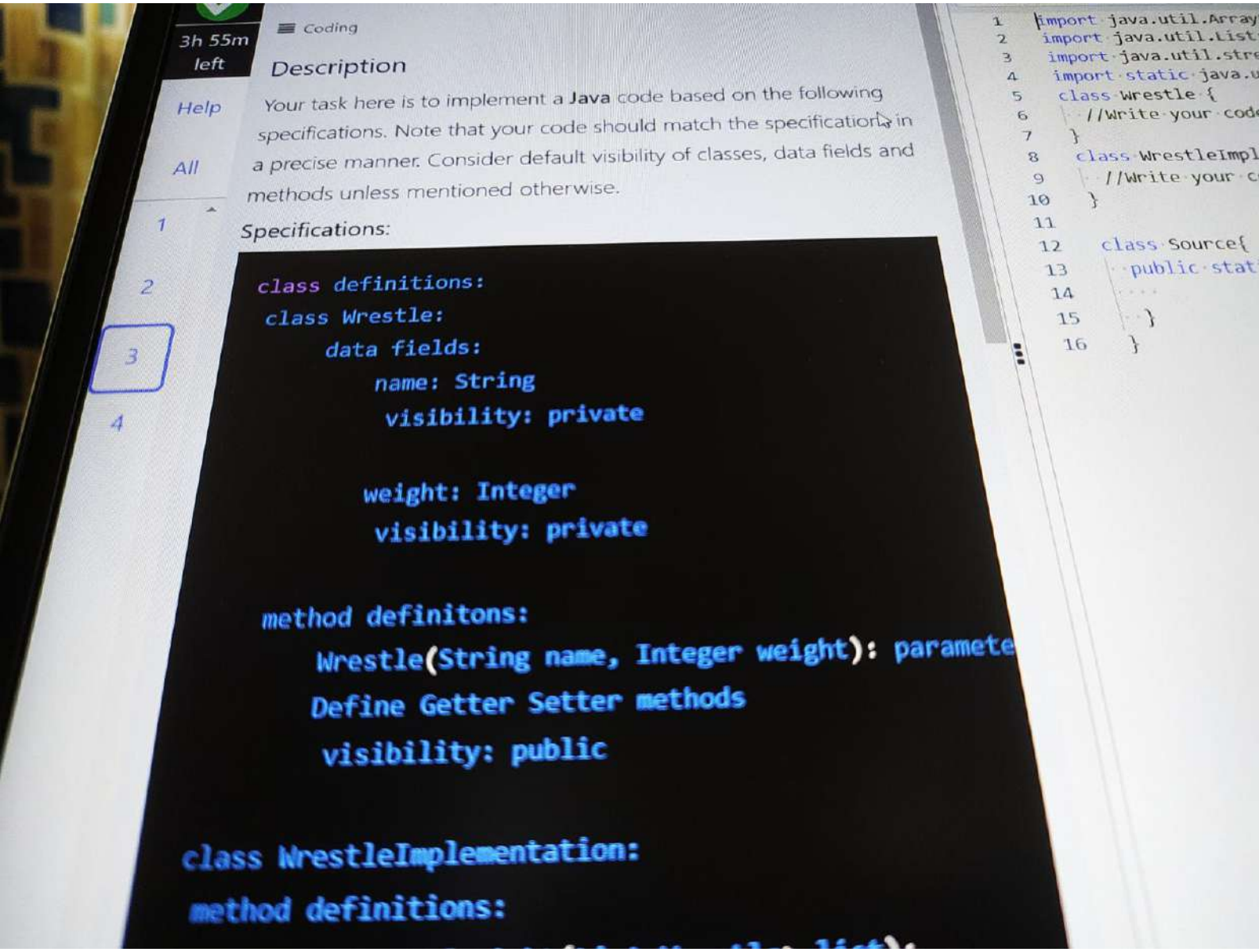
Autocomplete reconnecting...

```
36          if(!i.isInStock){
37              throw new StockBalanceException("item is out of stock");
38          }
39          else if(c.walletBalance < i.price){
40              throw new StockBalanceException("customer wallet balance is not sufficient");
41          }
42          else{
43              return "Order Successful";
44          }
45      }
46  }
47
48  class StockBalanceException extends Exception {
49      public StockBalanceException(String message){
50          super(message);
51      }
52      //Write Your Code Here..
53  }
54
55  public class Source {
56      public static void main(String args[] ) throws Exception {
57          /* Enter your code here. Read input from STDIN. Print output to STDOUT */
58      }
59  }
```

Autocomplete reconnecting...

RUN CODE

Test Results | Custom Input


3h 55m left
Coding
Description
Help
All
Your task here is to implement a Java code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.
Specifications:
1
2
3
4
class definitions:
class Wrestle:
data fields:
name: String
visibility: private
weight: Integer
visibility: private
method definitons:
Wrestle(String name, Integer weight): paramete
Define Getter Setter methods
visibility: public
class WrestleImplementation:
method definitions:
1 import java.util.Array
2 import java.util.List
3 import java.util.str
4 import static java.u
5 class Wrestle {
6 //Write your cod
7 }
8 class WrestleImpl
9 //Write your c
10 }
11
12 class Source{
13 public stat
14
15 }
16 }

3h 55m left

Help

All

1

2

3

4

```
class WrestleImplementation:
method definitions:
        getSumOfWeight(List<Wrestle> list):
            return type: int
            visibilty: public

        getWrestlerName(List<Wrestle> list):
            return type: List<String>
            visibilty: public

        getMaxWeight(List<Wrestle> list):
            return type: int
            visibilty: public
```

```
1  import java.util.Arrayl
2  import java.util.List;
3  import java.util.strea
4  import static java.ut
5  class Wrestle {
6      //Write your code
7  }
8  class WrestleImple
9      //Write your co
10 }
11
12 class Source{
13     public stati
14
15     }
16 }
```

Task:

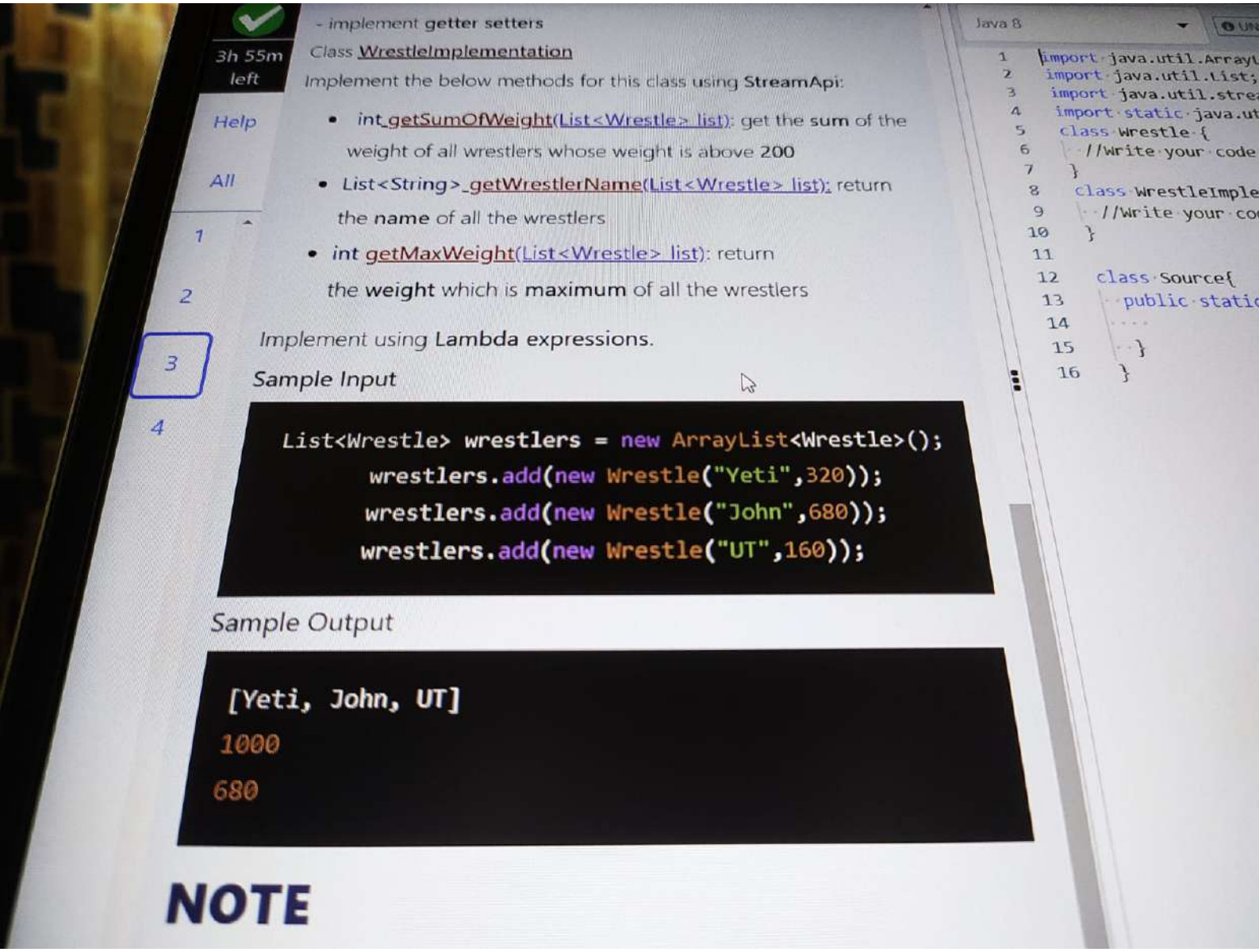
Class Wrestle

- define the String variable name
- define the Integer variable weight
- define a constructor
- implement getter setters

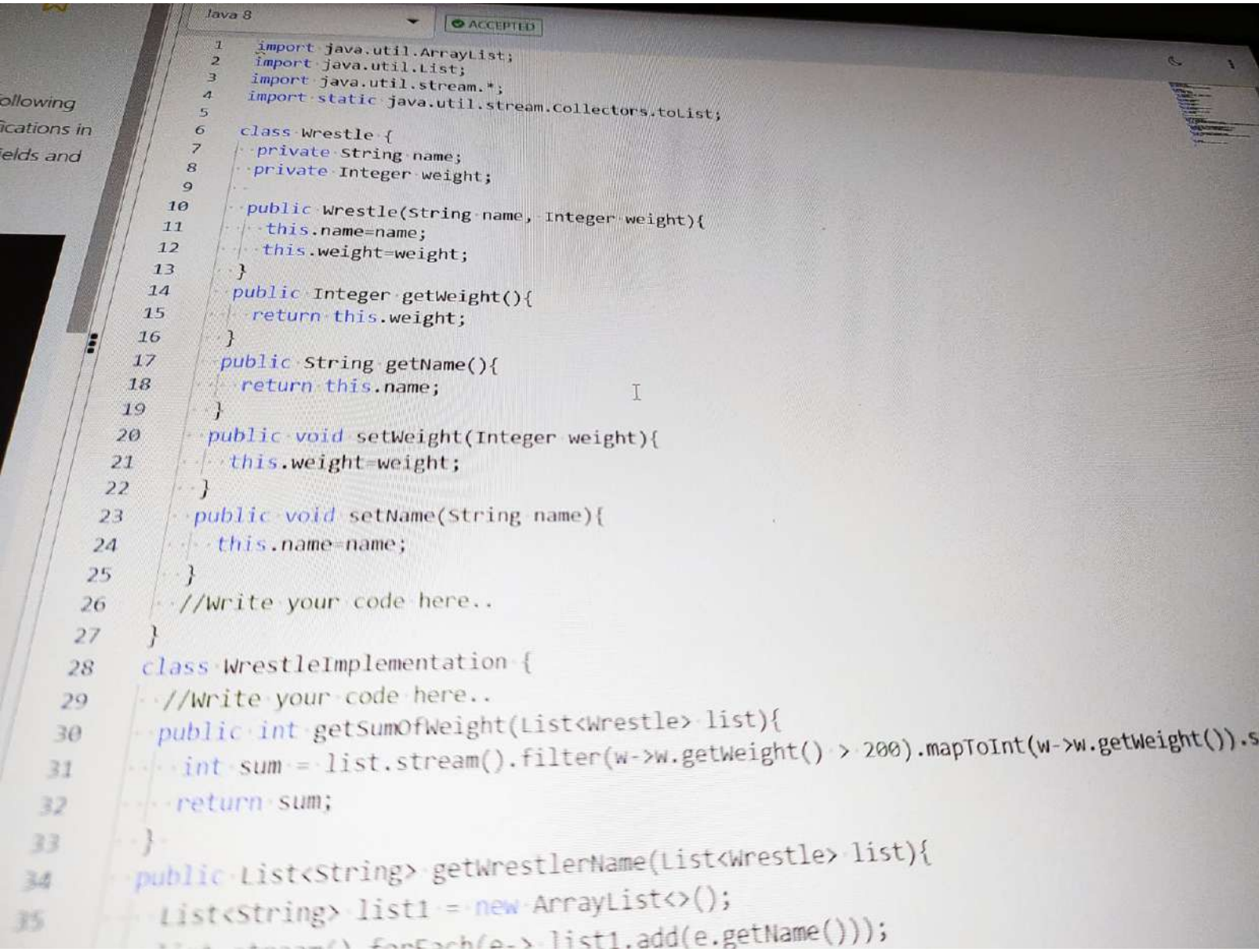
Class WrestleImplementation

Implement the below methods for this class using StreamApi:

- int getSumOfWeight(List<Wrestle> list): get the sum of the weight of all wrestlers whose weight is above 200


- implement getter setters
3h 55m left
Class WrestleImplementation
Implement the below methods for this class using StreamApi:
Help
All
int getSumOfWeight(List<Wrestle> list): get the sum of the weight of all wrestlers whose weight is above 200
List<String> getWrestlerName(List<Wrestle> list): return the name of all the wrestlers
int getMaxWeight(List<Wrestle> list): return the weight which is maximum of all the wrestlers
Implement using Lambda expressions.
Sample Input
List<Wrestle> wrestlers = new ArrayList<Wrestle>();
wrestlers.add(new Wrestle("Yeti",320));
wrestlers.add(new Wrestle("John",680));
wrestlers.add(new Wrestle("UT",160));
Sample Output
[Yeti, John, UT]
1000
680
Java 8
import java.util.Arrayl
import java.util.List;
import java.util.stre
import static java.ut
class Wrestle {
//Write your code
}
class WrestleImple
//Write your co
}
class Source{
public static
}
}

NOTE



Java 8 ACCEPTED

```
import java.util.ArrayList;
import java.util.List;
import java.util.stream.*;
import static java.util.stream.Collectors.toList;

class Wrestle {
    private String name;
    private Integer weight;

    public Wrestle(String name, Integer weight){
        this.name=name;
        this.weight=weight;
    }
    public Integer getWeight(){
        return this.weight;
    }
    public String getName(){
        return this.name;
    }
    public void setWeight(Integer weight){
        this.weight=weight;
    }
    public void setName(String name){
        this.name=name;
    }
    //Write your code here..
}
class WrestleImplementation {
    //Write your code here..
    public int getSumOfWeight(List<Wrestle> list){
        int sum = list.stream().filter(w->w.getWeight() > 200).mapToInt(w->w.getWeight()).s
        return sum;
    }
    public List<String> getWrestlerName(List<Wrestle> list){
        List<String> list1 = new ArrayList<>();
```

following
cifications in
fields and

```
    public String getName(){
        return this.name;
    }
    public void setWeight(Integer weight){
        this.weight=weight;
    }
    public void setName(String name){
        this.name=name;
    }
    //Write your code here..
}
class WrestleImplementation {
    //Write your code here..
    public int getSumOfWeight(List<Wrestle> list){
        int sum = list.stream().filter(w->w.getWeight() > 200).mapToInt(w->w.getWeight()).sum();
        return sum;
    }
    public List<String> getWrestlerName(List<Wrestle> list){
        List<String> list1 = new ArrayList<>();
        list.stream().forEach(e-> list1.add(e.getName()));
        return list1;
    }
    public int getMaxWeight(List<Wrestle> list){
        int max = list.stream().mapToInt(w->w.getWeight()).max().getAsInt();
        return max;
    }
}

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

    }
}
```

Autocomplete reconnecting...

RUN CO

3h 58m left

Help

All

1

2

3

4

Coding

Description

Danish has opened a seed bags selling shop. He wants to arrange the bags in the order in which a bag expires first. These bags have a twelve-digit code where:

- the first four characters are Batch Number.
- The next 8 digits represent the seed expiry date in **YYYYMMDD** format.

Generate a **Java** code to extract the batch number and expiry date from package code and validate it.

Validations:

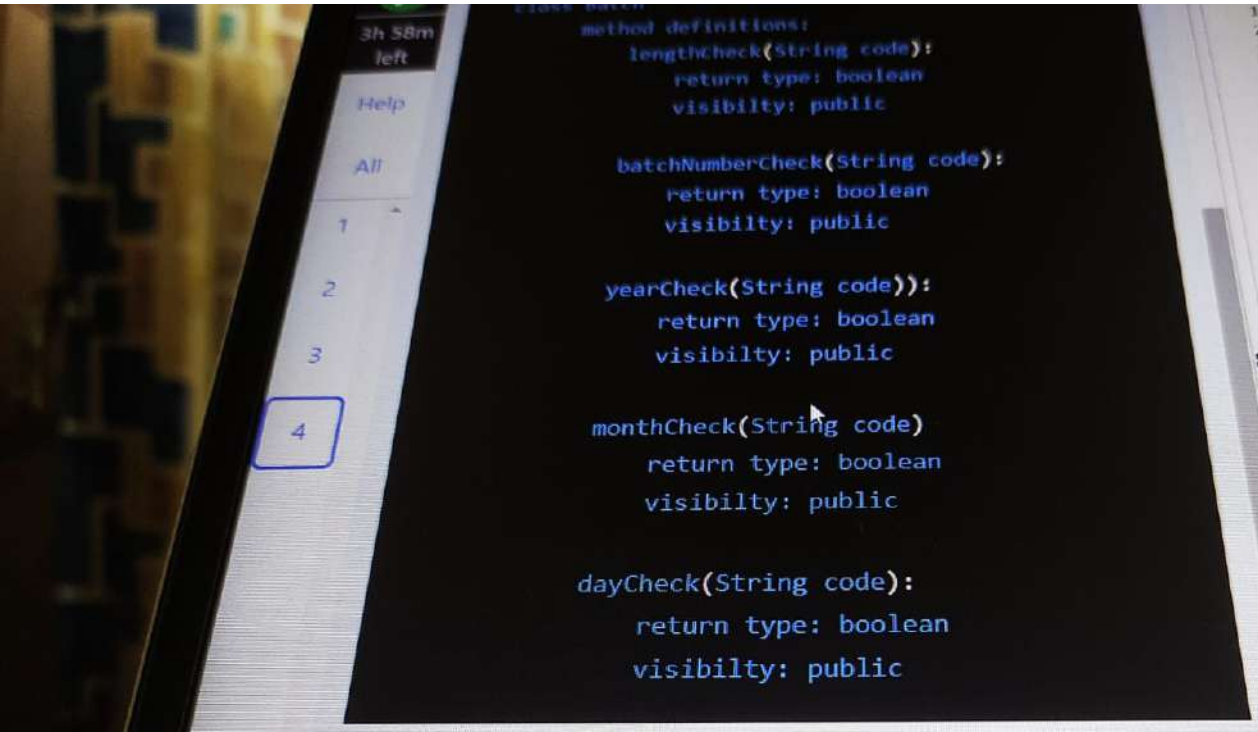
- Batch number is valid only if the first, second and fourth characters are alphabets in uppercase and the third character is a number.
- Year is valid only if it is between 2015 and 2020, both inclusive.
- Month is valid only if it is in between 1 and 12 (both inclusive).
- Day is valid only if it is between 1 and 31 (both inclusive).

Assumption:

- All characters in the input string will be in UPPER CASE.

```
class definitions:
```

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.SimpleDater
4  import java.util.Date;
5
6  class Batch{
7
8      public boolean lengt
9          //RETURN true if
10     }
11
12     public boolean bat
13         //RETURN true
14     }
15
16     public boolean
17         //Check if
18     }
19
20     public boole
21         //Check
22     }
23
24     public bo
25         //Che
26     }
27
28 }
29
30 class Sou
31     publi
32
33     }
34 }
```



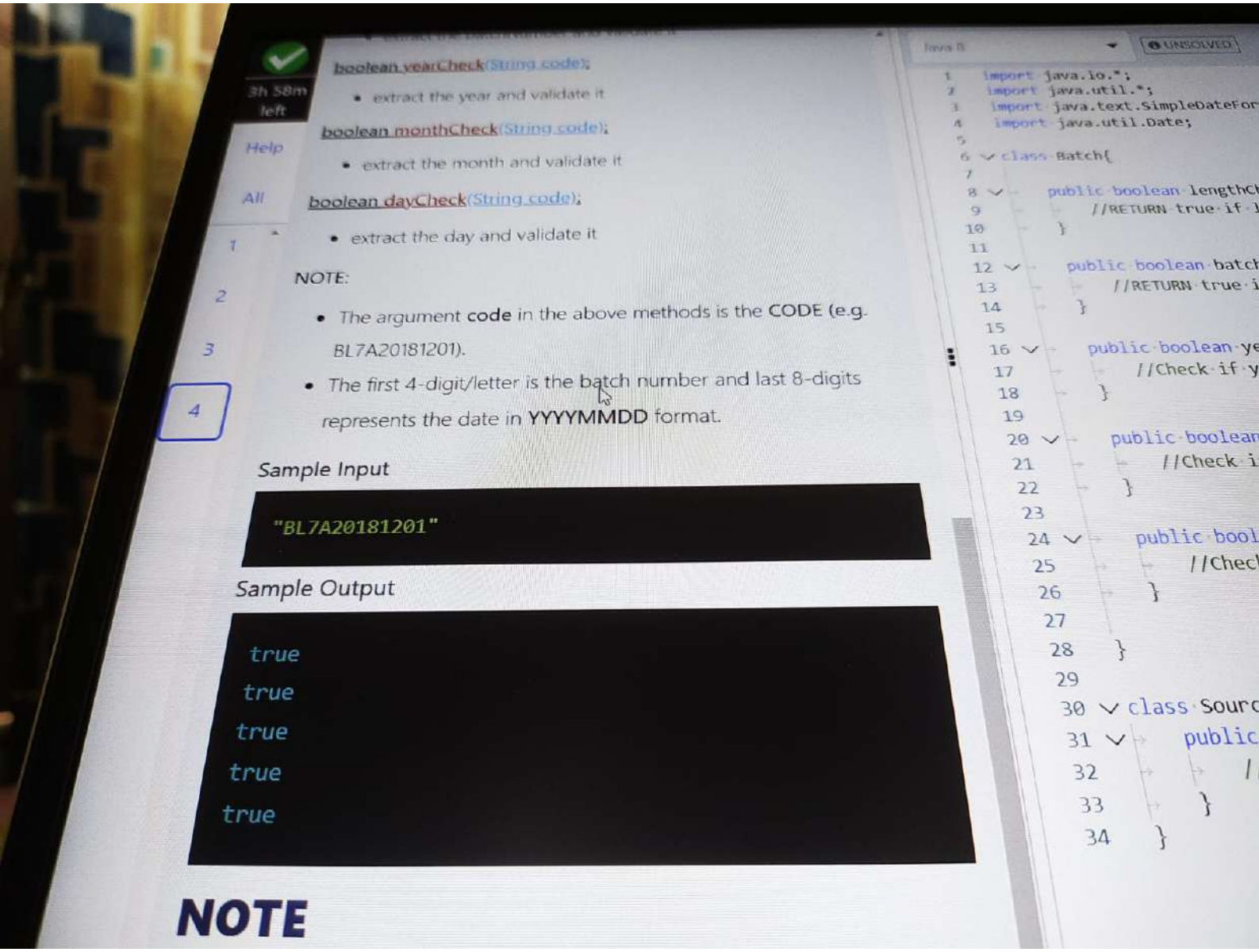
Methods to be Implemented:

boolean lengthCheck(String code);

- Compute the length of String **code** and return true if length is 12 else return false.

boolean batchNumberCheck(String code);

- extract the batchNumber and validate it


3h 58m left
Help
All
1
2
3
4
boolean yearCheck(String code);
extract the year and validate it
boolean monthCheck(String code);
extract the month and validate it
boolean dayCheck(String code);
extract the day and validate it
NOTE:
The argument code in the above methods is the CODE (e.g. BL7A20181201).
The first 4-digit/letter is the batch number and last 8-digits represents the date in YYYYMMDD format.
Sample Input
"BL7A20181201"
Sample Output
true
true
true
true
true
NOTE
Java 8
UNSOLVED
import java.io.*;
import java.util.*;
import java.text.SimpleDateFor
import java.util.Date;
class Batch{
public boolean lengthCh
//RETURN true if
public boolean batch
//RETURN true i
public boolean ye
//Check if y
public boolean
//Check i
public bool
//Chec
class Sourc
public
/
}

wants to arrange
hese bags have a

expiry date

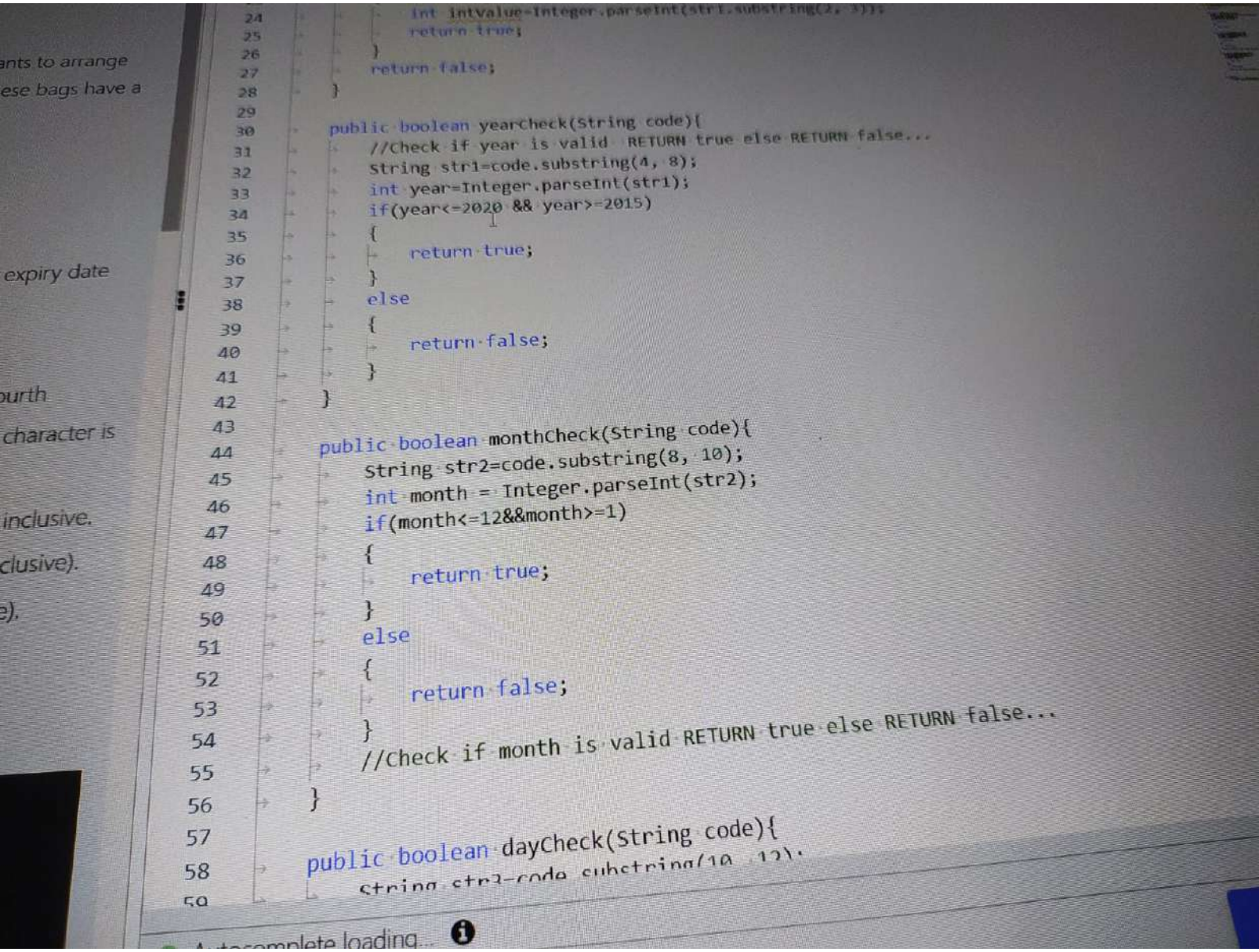
rth

haracter is

clusive.

sive).

```java
import java.io.*;
import java.util.*;
import java.text.SimpleDateFormat;
import java.util.Date;

class Batch{

    public boolean lengthCheck(String code){
        if(code.length()==12)
            return true;
            return false;
    }

    public boolean batchNumberCheck(String code){
        String str1=code.substring(0, 4);
        String check1=str1.substring(0, 1);
        String check2=str1.substring(1, 2);
        String check3=str1.substring(3,4);
        char x=check1.charAt(0);
        char y=check2.charAt(0);
        char z=check3.charAt(0);
        if(Character.isUpperCase(x)&& Character.isUpperCase(y) && Character.isUpperCase(z) )
        {
            int intValue=Integer.parseInt(str1.substring(2, 3));
            return true;
        }
        return false;
    }

    public boolean yearCheck(String code){
        //Check if year is valid  RETURN true else RETURN false...
        String str1=code.substring(4, 8);
        int year=Integer.parseInt(str1);
        if(year<=2020 && year>=2015)
        {
            return true;
        }
```



```
        int intValue=Integer.parseInt(str1.substring(2, 3));
        return true;
    }
    return false;
}

public boolean yearcheck(String code){
    //Check if year is valid  RETURN true else RETURN false...
    String str1=code.substring(4, 8);
    int year=Integer.parseInt(str1);
    if(year<=2020 && year>=2015)
    {
        return true;
    }
    else
    {
        return false;
    }
}

public boolean monthCheck(String code){
    String str2=code.substring(8, 10);
    int month = Integer.parseInt(str2);
    if(month<=12&&month>=1)
    {
        return true;
    }
    else
    {
        return false;
    }
    //Check if month is valid RETURN true else RETURN false...
}

public boolean dayCheck(String code){
    String str3=code.substring(10, 12);
```

Autocomplete loading...

op. He wants to arrange
s first. These bags have a

ber
piry date

ber and expiry date

d and fourth
e third character is

), both inclusive.
both inclusive).
nclusive).

CASE.

```
    public boolean monthCheck(String code){
        String str2=code.substring(8, 10);
        int month = Integer.parseInt(str2);
        if(month<=12&&month>=1)
        {
            return true;
        }
        else
        {
            return false;
        }
        //Check if month is valid RETURN true else RETURN false...
    }

    public boolean dayCheck(String code){
        String str3=code.substring(10, 12);
        int date=Integer.parseInt(str3);
        if(date<=31&&date>=1)
        {
            return true;
        }
        else
        {
            return false;
        }
        //Check if day is valid RETURN true or else RETURN false...
    }

}

class Source{
    public static void main(String[] args) {
        //Don't touch this or else your program might not run...
    }
}
```

Autocomplete reconnecting...

class StringMethods

- Implement the below methods for this class:

  - int convertToInt(StringPlay sp, String str): Convert the string str to int, return the int value and assign the value to suitable sp variable(convert). All the strings will contain only numbers.
  - Example: str = "123" then resultant is 123
  - int getMax(StringPlay sp, String str, char ch): Return the total number of char ch present in string str and assign the value to sp variable max and return the same value.
  - Example: str = "This is good", ch = 'o' then resultant value = 2
  - Example: str = "doselect Et le", ch='e' then resultant value = 3

Sample Input

```
StringMethods sm = new StringMethods();
StringPlay sp = new StringPlay();
sm.getMax(sp,"fefefef",'e')
```

```
15  class StringMethods{
16      public int convertToInt(StringPlay sp, String str){
17          int num=Integer.parseInt(str);
18          int convert=num;
19          return convert;
20      }
21      public int getMax(StringPlay sp, String str, char ch){
22          int k=0;
23  for(int i=0;i<str.length()-1;i++){
24          if(str.charAt(i)==ch){
25              k++;
26          }
27  }
28  int max=k;
29  return max;
30  }
```

Autocomplete reconnecting...

Ln 28, Col 10 Java 8

Test Results | Custom Input | RUN CODE | SUBMIT

Test_getMax1

Test_getMax2

Test_getMax2

Test_StringPlay

Passed

Description

Testcase passed!

2.31s	0s	1MB
Eval Time	CPU	Memory

```
13    }
14
15    class StringMethods{
16        public int convertToInt(StringPlay sp, String str){
17            int num=Integer.parseInt(str);
18            return num;
19        }
20        public int getMax(StringPlay sp, String str, char ch){
21            int k=0;
22    for(int i=0;i<str.length()-1;i++){
23            if(str.charAt(i)==ch){
24                k++;
25            }
26    }
27    return k;
28
29    }
30    }
31    public class Source {
32        public static void main(String args[] ) throws Exception {
33            /* Enter your code here. Read input from STDIN. Print output to STDOUT */
34        }
35    }
```

Autocomplete reconnecting...

- Implement the below methods for this class.

- int convertToInt(StringPlay sp, String str): Convert the string str to int, return the int value and assign the value to suitable sp variable(convert). All the strings will contain only numbers.
- Example: str = "123" then resultant is 123.
- int getMax(StringPlay sp, String str, char ch): Return the total number of char ch present in string str and assign the value to sp variable max and return the same value.
- Example: str = "This is good" , ch = 'o' then resultant value = 2
- Example: str = "doselect Et le", ch='e' then resultant value = 3

Sample Input

```
StringMethods sm = new StringMethods();
StringPlay sp = new StringPlay();
sm.getMax(sp,"fgfgfgf",'g')
sm.convertToInt(sp,"123")
```

Sample Output

```
3
123
```

NOTE

Java 8

```
1  > import java.i
10
11   class StringM
12     ..
13   }
14
15   public class
16     public s
17       /* P
18     }
19   }
```

Autocomplete

1h 22m left

Help

All

1

2

3

4

```
StringPlay(): Define an empty constructor
class StringMethods:
 convertToInt(StringPlay sp, String str):
  visibility: public
  return type:int
 getMax(StringPlay sp, String str, char ch):
  visibility: public
  return type:int
```

Task:

class StringPlay

- Implement StringPlay class according to the above specifications

class StringMethods

- Implement the below methods for this class:

- int convertToInt(StringPlay sp, String str): Convert the string str to int, return the int value and assign the value to suitable sp variable(convert). All the strings will contain only numbers.
- Example: str = "123" then resultant is 123
- int getMax(StringPlay sp, String str, char ch): Return the total number of char ch present in string str and assign the value to sp variable max and return the same value.
- Example: str = "This is good" , ch = 'o' then resultant

Java 8

```
1  > impor
10
11   class
12
13   {
14
15   publ
16
17
18
19   }
```

Autoco

Test Re

1h 22m left

Help

All

1

2

3

4

4. Harry's Assignment

Coding

Description

Harry has recently learned about strings in his programming classes. He decided to create some interesting strings using the basic concepts.

Help Harry!

Your task here is to implement a JAVA code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise. All the methods that you are implementing should be non-static.

Specification:

```
class definitions:
 class StringPlay:
  data fields:
   int convert
   int max;
  StringPlay(): Define an empty constructor wit

 class StringMethods:
  convertToInt(StringPlay sp, String str):
   visibility: public
   return type:int
  getMax(StringPlay sp, String str, char ch):
   visibility: public
```

```
    }
}
class ExceptionCheck{
//Implement the ExceptionCheck methods as per specification.
    public String validateEmployee(Employee emp) throws Exception{
        try {
            if(emp.employeeName == null || emp.employeeName.length()<3)
                throw new InvalidEmployeeException("Employee name invalid");
            else if(emp.employeeID == null || emp.employeeID<=100)
                throw new InvalidEmployeeException("Employee id invalid");
            if(emp.employeeName.length()>=3 || emp.employeeID>100)
                emp.status = "success";
        } catch(InvalidEmployeeException e) {
            emp.status = e.getMessage();
            throw new InvalidEmployeeException(e.getMessage());
        } catch(Exception e) {
            throw new Exception("unknown error occured");
        }
        return emp.status;
    }
}

public class Source {
    public static void main(String args[] ) throws Exception {
        //Implement main() to check your program..
    }
}
```

Task:

Class Student

- define the String variable studentName
- define the Integer variable studentRoll
- define the Integer variable studentScore
- define a parameterized constructor
- define getter setter

Class Implementation

Implement the below methods for this class using StreamAPI:

- List<Student> sortByScore(List<Student> student): Method to return sorted list by score (Refer to sample Output for clarity)
- Long getScoreCountAbove35(List<Student> student): Method to return the count of students whose score is above 35

Implement using Lambda expressions.

Note

- Following has been done for you:
- String toString() method, it's part of code stub, don't edit it else your *test-cases might fail*

Sample Input

```
List<Student> s = new ArrayList<>();
s.add(new Student("Ayaan", 12, 99));
s.add(new Student("Jaenny", 14, 29));
s.add(new Student("Shaeryl", 17, 89));
s.add(new Student("Sarah", 19, 55));
```

mple Output

```
List<Student> s = [illegible]<>();
s.[illegible]([illegible]([illegible], [illegible], [illegible]));
s.[illegible]([illegible]([illegible], [illegible], [illegible]));
s.[illegible]([illegible]([illegible], [illegible], [illegible]));
s.[illegible]([illegible]("Sarah", 19, 55));
```

Sample Output

```
[Student{name='Jaenny', roll=14, score=29},S
3
```

NOTE

- You can make suitable function calls and use RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN ISS

```
List<Student> s = [illegible] <>();
s. [illegible] ( [illegible] ( [illegible] , [illegible] , [illegible] ));
s. [illegible] ( [illegible] ( [illegible] , [illegible] , [illegible] ));
s. [illegible] ( [illegible] ( [illegible] , [illegible] , [illegible] ));
s. [illegible] ( [illegible] ("Sarah", 19, 55));
```

Sample Output

```
[Student{name='Jaenny', roll=14, score=29},S
3
```

NOTE

- You can make suitable function calls and use RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN ISS

```
List<Student> s = [illegible]<>();
s.[illegible]([illegible]([illegible], [illegible], [illegible]));
s.[illegible]([illegible]([illegible], [illegible], [illegible]));
s.[illegible]([illegible]([illegible], [illegible], [illegible]));
s.[illegible]([illegible] Student("Sarah", 19, 55));
```

Sample Output

```
[Student{name='Jaenny', roll=14, score=29}, S
3
```

NOTE

- You can make suitable function calls and use RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN ISS

All

```
List<Student> s =                 <>();
s.  (         (        ,     ,     ));
s.  (         (        ,     ,     ));
s.  (         (        ,     ,     ));
s.  (    Student("Sarah", 19, 55));
```

Sample Output

```
[Student{name='Jaenny', roll=14, score=29},S
3
```

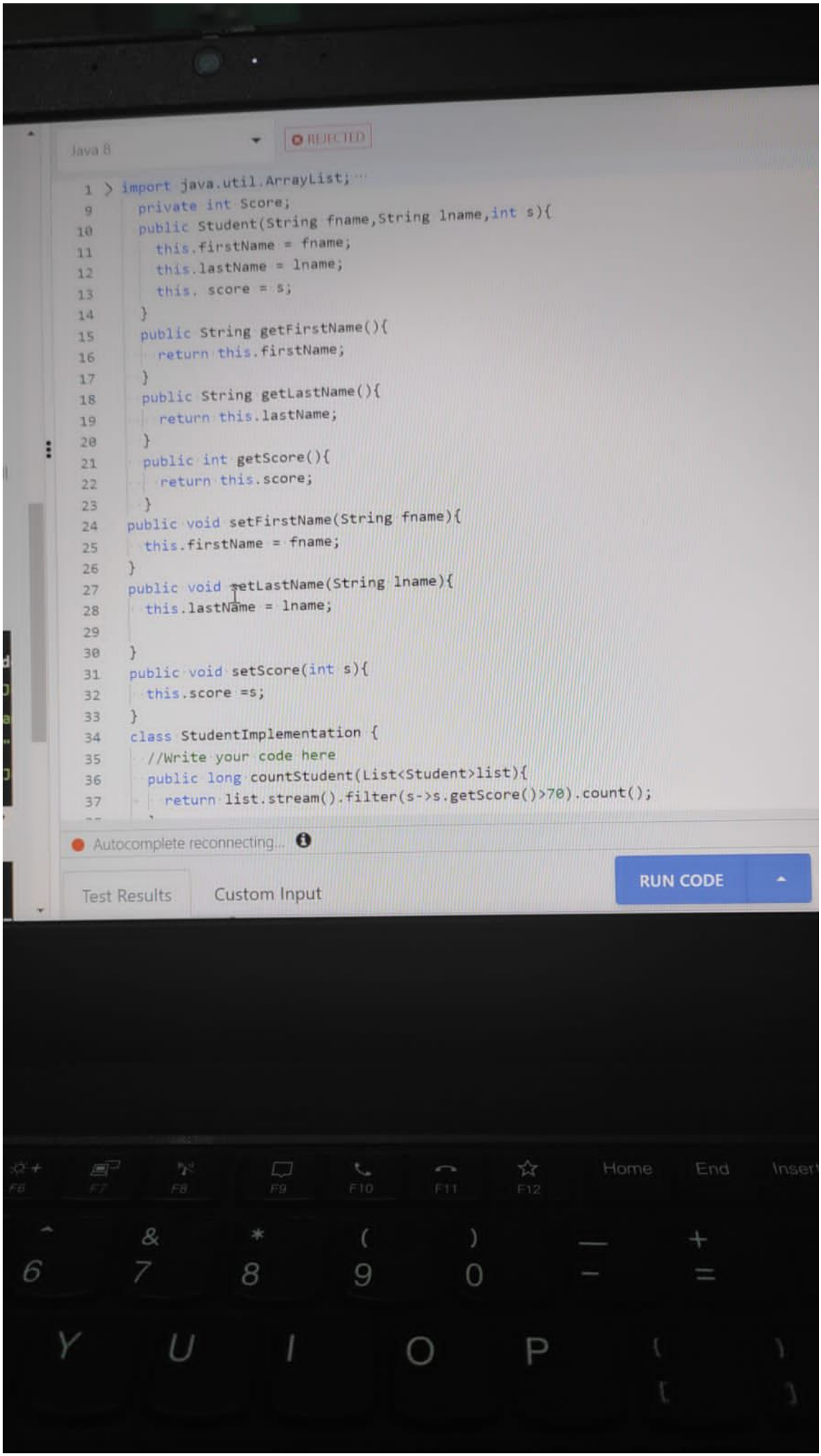
NOTE

- You can make suitable function calls and use RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN ISS


Java 8
REJECTED
1 > import java.util.ArrayList;
9 private int Score;
10 public Student(String fname,String lname,int s){
11 this.firstName = fname;
12 this.lastName = lname;
13 this. score = s;
14 }
15 public String getFirstName(){
16 return this.firstName;
17 }
18 public String getLastName(){
19 return this.lastName;
20 }
21 public int getScore(){
22 return this.score;
23 }
24 public void setFirstName(String fname){
25 this.firstName = fname;
26 }
27 public void setLastName(String lname){
28 this.lastName = lname;
29
30 }
31 public void setScore(int s){
32 this.score =s;
33 }
34 class StudentImplementation {
35 //Write your code here
36 public long countStudent(List<Student>list){
37 return list.stream().filter(s->s.getScore()>70).count();
Autocomplete reconnecting...
Test Results
Custom Input
RUN CODE
Home
End
F8
F7
F8
F9
F10
F11
F12
6
7
8
9
0
Y
U
I
O
P

3. Score Card

Coding

Description

Your task here is to implement a Java code based on the following specifications. Note that your code should match the specifications i a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

Specifications:

```
Class definitions:
  Class Student:
    Data fields:
      studentName: String
      studentRoll: Integer
      studentScore: Integer

  Method definitions:
    Define a parameterized constructor
      Student(String studentName, Integer studentRoll,
        visibility: public
    Implement getter setter methods
      visibility: public

  class Implementation:
      sortByScore(List<Student> student):
          Return type: List<Student>
          Visibility: public

      getScoreCountAbove35(List<Student> student):
          Return type: Long
          Visibility: public
```

Task:

Class Student

- define the String variable studentName
- define the Integer variable studentRoll
- define the Integer variable studentScore
- define a parameterized constructor
- define getter setter

Class Implementation

Implement the below methods for this class using StreamAPI:

- List<Student> sortByScore(List<Student> student): Method to return sorted list by score (Refer to sample Output for clarity)
- Long getScoreCountAbove35(List<Student> student): Method to return the count of students whose score is above 35

Implement using Lambda expressions.

Note

- Following has been done for you:
- String toString() method, it's part of code stub, don't edit it else your *test-cases might fail*

Sample Input

```
List<Student> s = new ArrayList<>();
s.add(new Student("Ayaan", 12, 99));
s.add(new Student("Jaenny", 14, 29));
s.add(new Student("Shaeryl", 17, 89));
s.add(new Student("Sarah", 19, 55));
```

mple Output

owing

ations in

```java
import java.text.*;
import java.math.*;
import java.util.regex.*;
import java.util.*;
import java.io.*;
import java.lang.*;
import java.io.FileReader;
import java.io.IOException;

class Employee {
    String employeeName;
    Integer employeeID;
    String status;
    public Employee(Integer id, String name) {
        this.employeeName = name;
        this.employeeID = id;
    }
}
class InvalidEmployeeException extends Exception{
    public InvalidEmployeeException(String msg) {
        super(msg);
    }
}
class ExceptionCheck{
//Implement the ExceptionCheck methods as per specification.
validateEmployee(Employee emp) throws Exception{
```

39m 16s left

Help

All

1

2

3

4

1. List of Operations

Coding

Description

Your task here is to implement a **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

Specifications:

```
class definitions:
 class ArrayListOps:
 method definitions:
  makeArrayListInt(int n): Method to create
      return type: ArrayList<Integer>
      visibilty: public

  reverseList(ArrayList<Integer> list): Met
      return type: ArrayList<Integer>
      visibilty: public

  changeList(ArrayList<Integer> list, int m
      return type: ArrayList<Integer>
      visibilty: public
```

Task:

Java 8

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

Autoc

Test R

Esc FnLock F1 F2 F3 F4 F5

39m 2s left

Help

All

1

2

3

4

```
makeArrayListInt(int n): Method to create
    return type: ArrayList<Integer>
    visibilty: public

reverseList(ArrayList<Integer> list): Met
    return type: ArrayList<Integer>
    visibilty: public

changeList(ArrayList<Integer> list, int m
    return type: ArrayList<Integer>
    visibilty: public
```

Task:

Your task is to create a class **ArrayListOps** and implement the following:

1. **makeArrayListInt(int n)**: Method to create an **ArrayList** with number of elements as **n** and ***set*** *elements* to **0**.
    - If number of elements **n** is 4 , then the method should return a list containing **[0,0,0,0]**
2. **reverseList(ArrayList<Integer> list)**: Method to ***Reverse*** **list**
3. **changeList(ArrayList<Integer> list, int m, int n)**: Method to change all **occurences** of **m** to **n** in **list**

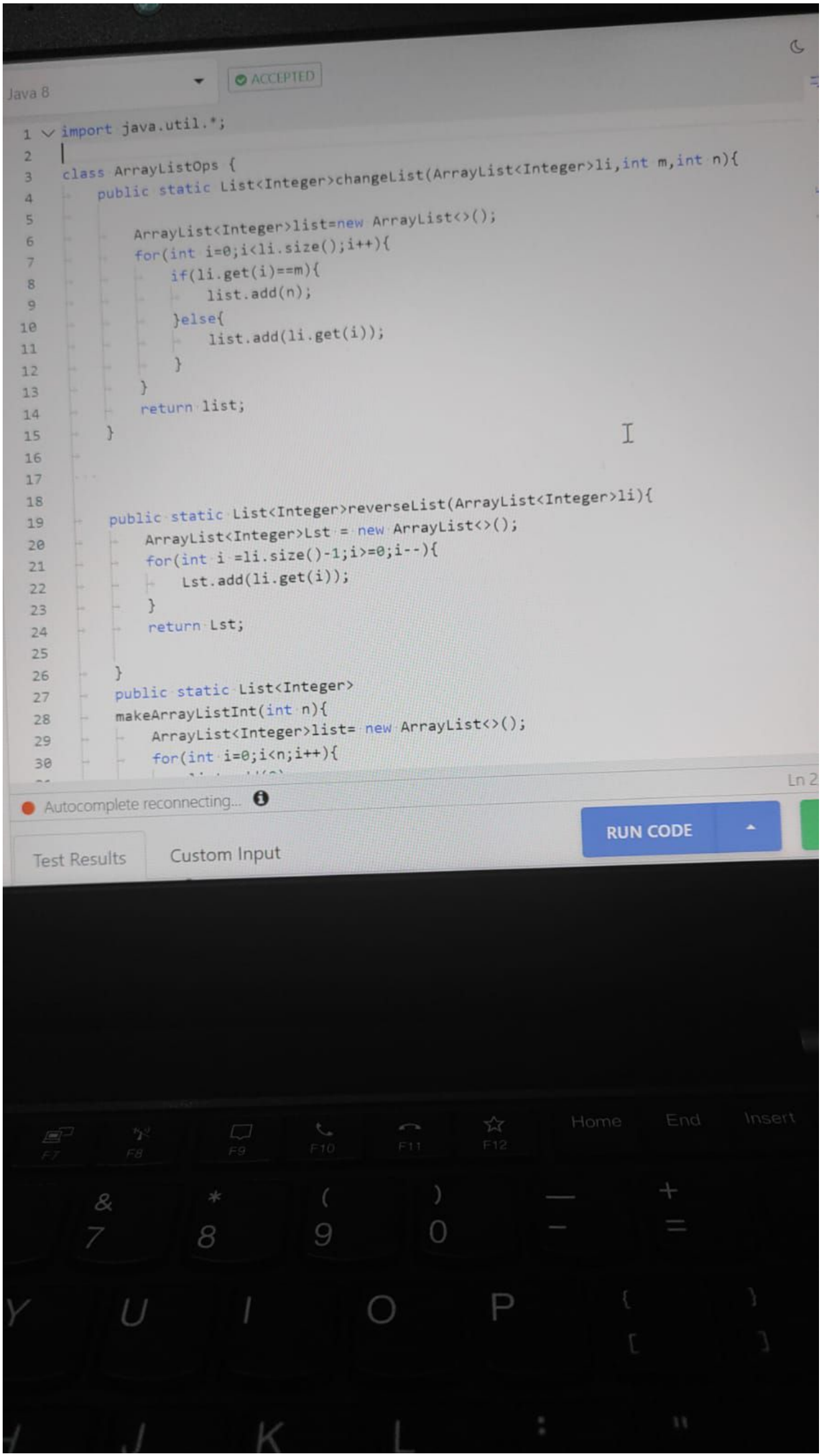
Important:

Java 8

1 > i
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

Auto

Test R



Java 8

ACCEPTED

```java
import java.util.*;

class ArrayListOps {
    public static List<Integer>changeList(ArrayList<Integer>li,int m,int n){

        ArrayList<Integer>list=new ArrayList<>();
        for(int i=0;i<li.size();i++){
            if(li.get(i)==m){
                list.add(n);
            }else{
                list.add(li.get(i));
            }
        }
        return list;
    }


    public static List<Integer>reverseList(ArrayList<Integer>li){
        ArrayList<Integer>Lst = new ArrayList<>();
        for(int i =li.size()-1;i>=0;i--){
            Lst.add(li.get(i));
        }
        return Lst;

    }
    public static List<Integer>
    makeArrayListInt(int n){
        ArrayList<Integer>list= new ArrayList<>();
        for(int i=0;i<n;i++){
```

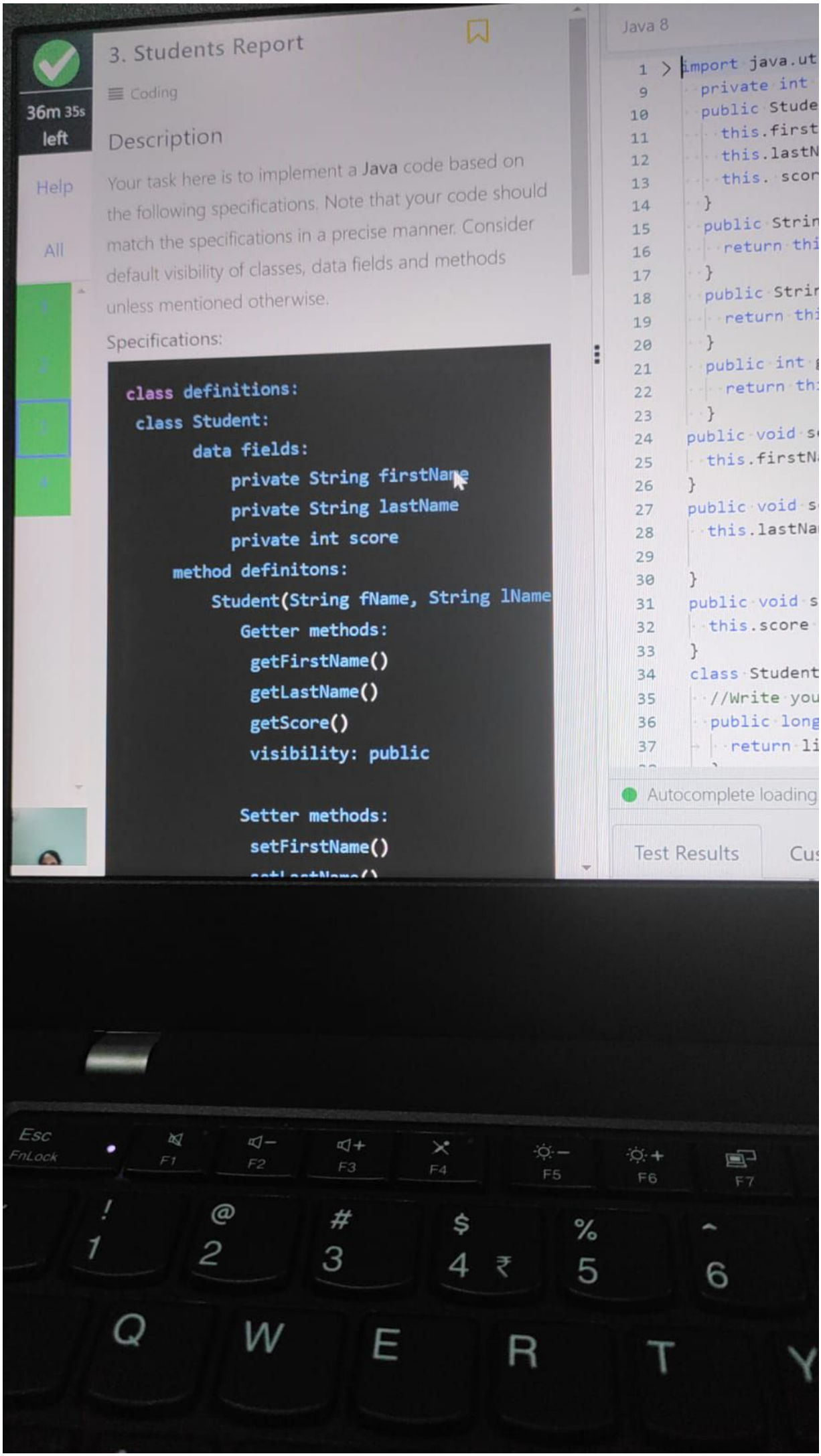
Ln 2

Autocomplete reconnecting...

Test Results

Custom Input

RUN CODE


36m 35s left
Help
All
3. Students Report
Coding
Description
Your task here is to implement a Java code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.
Specifications:
class definitions:
class Student:
data fields:
private String firstName
private String lastName
private int score
method definitons:
Student(String fName, String lName
Getter methods:
getFirstName()
getLastName()
getScore()
visibility: public
Setter methods:
setFirstName()
Java 8
Autocomplete loading
Test Results

37m 5s left

Help

All

- Define a catch block for handling the InvalidEmployeeException, assign exception message to employee.status and rethrow the same InvalidEmployeeException object
- Define a generic Exception handler catch block that handles all the other exceptions by rethrowing a new Exception object with message "unknown error occured"
- Return employee.status

class InvalidEmployeeException

- define custom exception class InvalidEmployeeException by extending the Exception class.
- define a parameterized constructor with a String argument to pass the message to the super class.

NOTE

- You can make suitable function calls and use RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN ISSUE

Java 8

1 > i
9
10
11
12 c
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

Auto

Test F

Esc
FnLock

38m 52s left

Help

All

- If number of elements n is 4 , then the method should return a list containing [0,0,0,0]

2. reverseList(ArrayList<Integer> list): Method to *Reverse* list

3. changeList(ArrayList<Integer> list, int m, int n): Method to change all occurences of m to n in list

Important:

- To check your program, you can use the main() method (in Source class) given in the stub. You can make suitable function calls and use RUN CODE button to check your main() function output.

Sample Input

```
ArrayList<Integer> list = new ArrayList<Int
n = 4(method makeArrayListInt)
m = 28, n = 20(method changeList)
```

Sample Output

```
[0, 0, 0, 0]
[12, 10, 28, 33, 25, 10]
[12, 10, 20, 33, 25, 10]
```

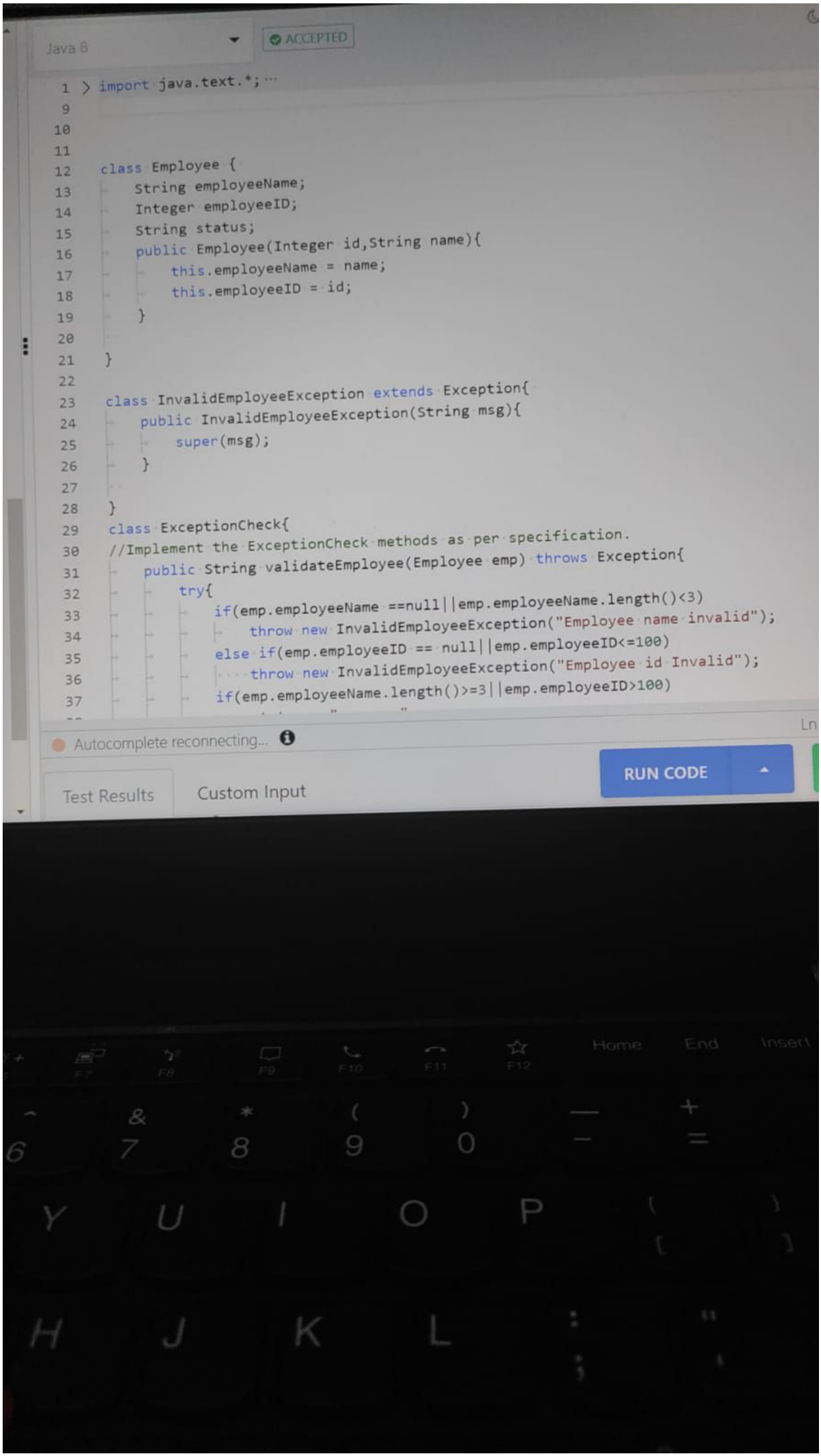
NOTE:

Java 8

```
1  > import
9
10
11
12
13
14
15      }
16
17
18
19      p
20
21
22
23
24
25
26      }
27      p
28      n
29
30
31
32
33
34
35
36
37  }
```

Autocomp

Test Result


Java 8
ACCEPTED
1 > import java.text.*;...
9
10
11
12 class Employee {
13 String employeeName;
14 Integer employeeID;
15 String status;
16 public Employee(Integer id,String name){
17 this.employeeName = name;
18 this.employeeID = id;
19 }
20
21 }
22
23 class InvalidEmployeeException extends Exception{
24 public InvalidEmployeeException(String msg){
25 super(msg);
26 }
27
28 }
29 class ExceptionCheck{
30 //Implement the ExceptionCheck methods as per specification.
31 public String validateEmployee(Employee emp) throws Exception{
32 try{
33 if(emp.employeeName ==null||emp.employeeName.length()<3)
34 throw new InvalidEmployeeException("Employee name invalid");
35 else if(emp.employeeID == null||emp.employeeID<=100)
36 throw new InvalidEmployeeException("Employee id Invalid");
37 if(emp.employeeName.length()>=3||emp.employeeID>100)
Autocomplete reconnecting...
Test Results
Custom Input
RUN CODE
Home
End
Insert

2. Catch Me If You Can

Coding

Description

Your task here is to implement a **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner.

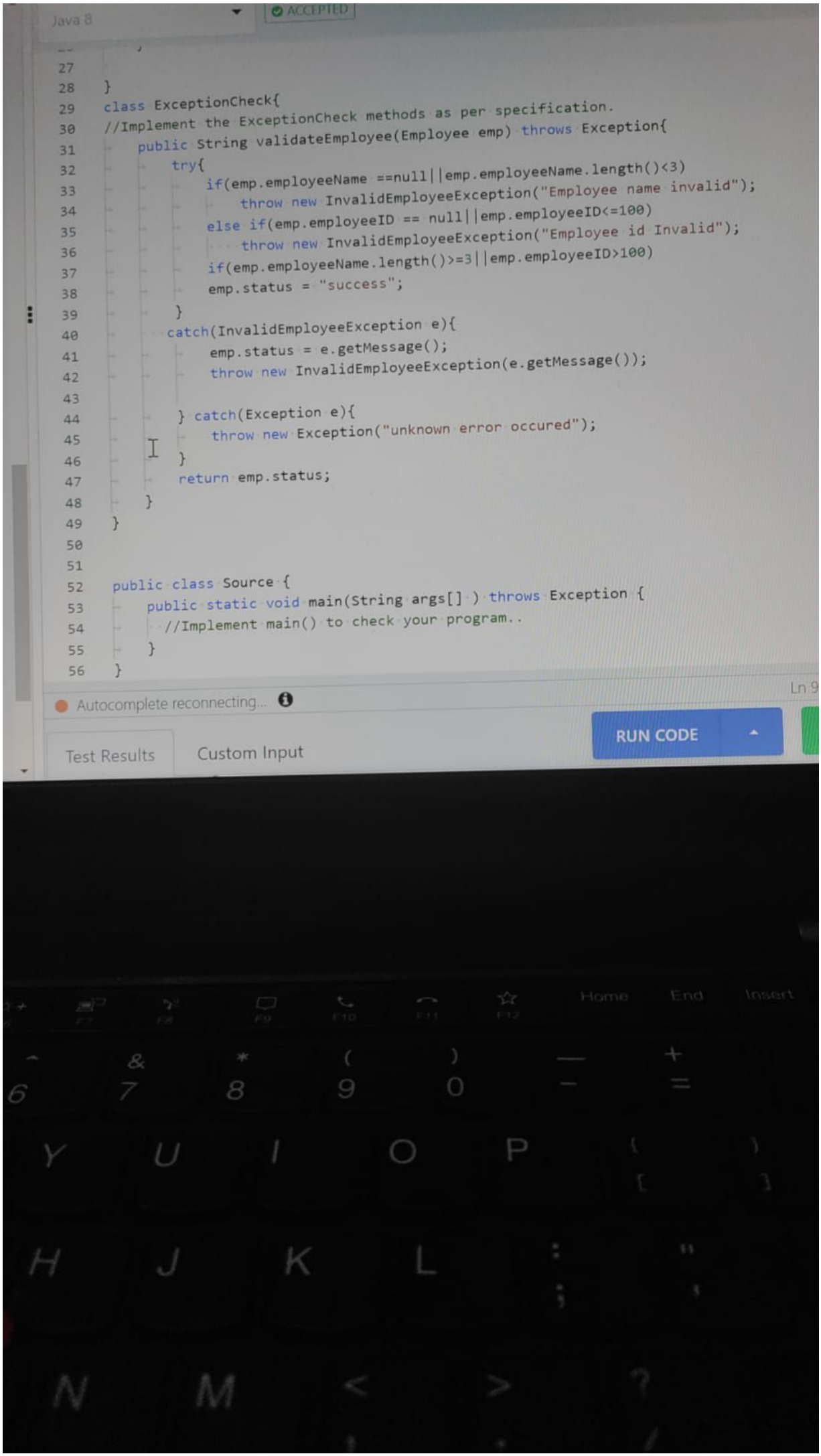
Consider **default visibility** of class unless mentioned otherwise.

```
class definitions:clas Employee:data fields
   employeeName: String
   employeeID: Integer
   status: String
  method definition:
   Employee(Integer id, String name):constr
     visibility: public

 class InvalidEmployeeException:
  method definitions:
    InvalidEmployeeException(String msg):
     visibility: public

class ExceptionCheck:
   validateEmployee(Employee employee) thro
    Visibility:public
    Return type: String
```

```
1  > import ja
9
10
11
12  class Emp
13      Strir
14      Integ
15      Strir
16      publi
17          t
18          t
19      }
20
21  }
22
23  class Inv
24      publ
25          s
26      }
27
28  }
29  class Ex
30  //Implem
31      publ
32          t
33          t
34          t
35          t
36          t
37          t
```

Autocomplete lo

Test Results



Java 8

ACCEPTED

```
27
28  }
29  class ExceptionCheck{
30  //Implement the ExceptionCheck methods as per specification.
31      public String validateEmployee(Employee emp) throws Exception{
32          try{
33              if(emp.employeeName ==null||emp.employeeName.length()<3)
34                  throw new InvalidEmployeeException("Employee name invalid");
35              else if(emp.employeeID == null||emp.employeeID<=100)
36                  throw new InvalidEmployeeException("Employee id Invalid");
37              if(emp.employeeName.length()>=3||emp.employeeID>100)
38              emp.status = "success";
39          }
40          catch(InvalidEmployeeException e){
41              emp.status = e.getMessage();
42              throw new InvalidEmployeeException(e.getMessage());
43
44          } catch(Exception e){
45              throw new Exception("unknown error occured");
46          }
47          return emp.status;
48      }
49  }
50
51
52  public class Source {
53      public static void main(String args[] ) throws Exception {
54          //Implement main() to check your program..
55      }
56  }
```

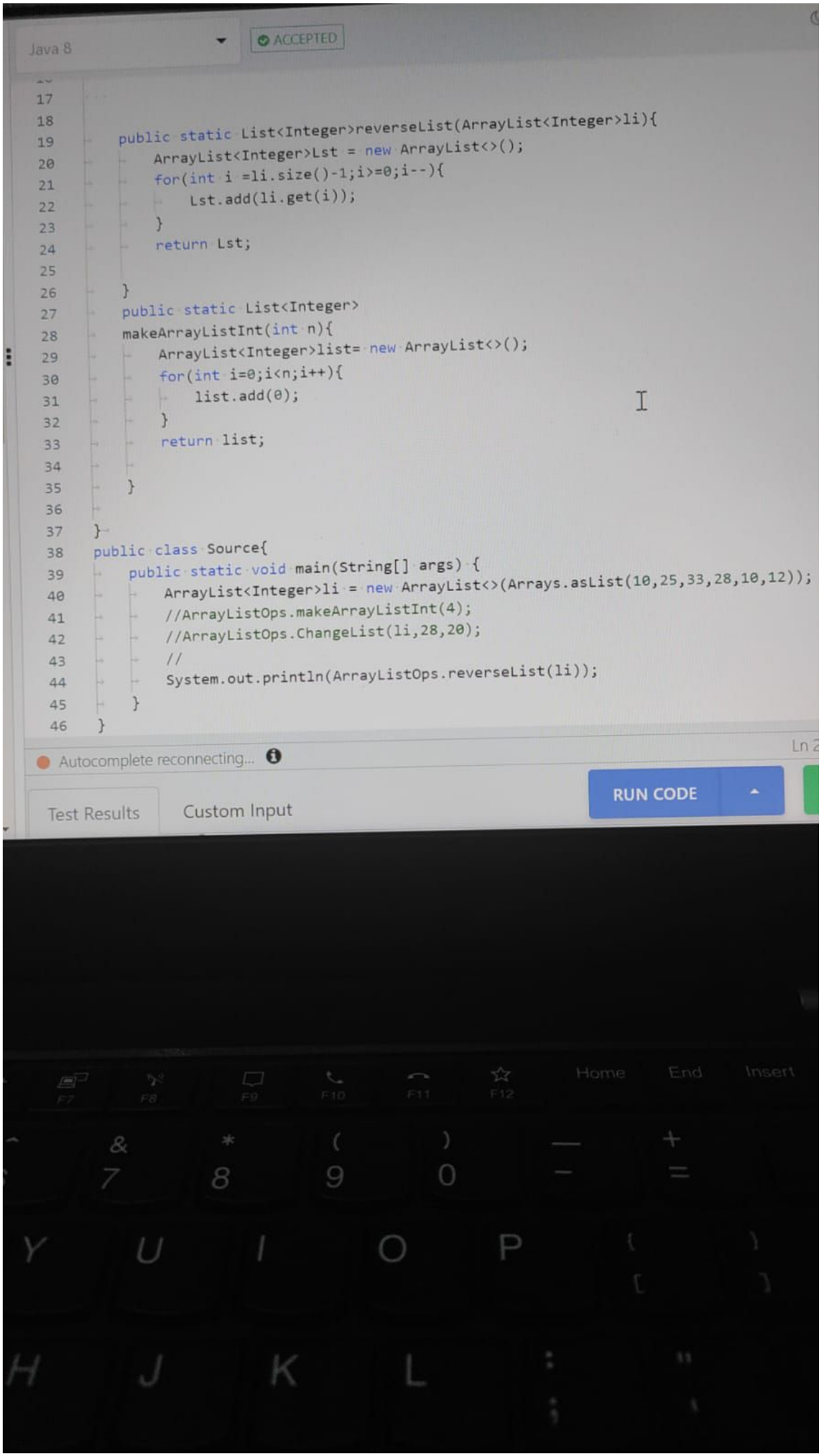
Autocomplete reconnecting...

Ln 9

Test Results

Custom Input

RUN CODE



Java 8

ACCEPTED

```
17
18
19      public static List<Integer>reverseList(ArrayList<Integer>li){
20          ArrayList<Integer>Lst = new ArrayList<>();
21          for(int i =li.size()-1;i>=0;i--){
22              Lst.add(li.get(i));
23          }
24          return Lst;
25
26      }
27      public static List<Integer>
28      makeArrayListInt(int n){
29          ArrayList<Integer>list= new ArrayList<>();
30          for(int i=0;i<n;i++){
31              list.add(0);
32          }
33          return list;
34
35      }
36
37  }
38  public class Source{
39      public static void main(String[] args) {
40          ArrayList<Integer>li = new ArrayList<>(Arrays.asList(10,25,33,28,10,12));
41          //ArrayListOps.makeArrayListInt(4);
42          //ArrayListOps.ChangeList(li,28,20);
43          //
44          System.out.println(ArrayListOps.reverseList(li));
45      }
46  }
```

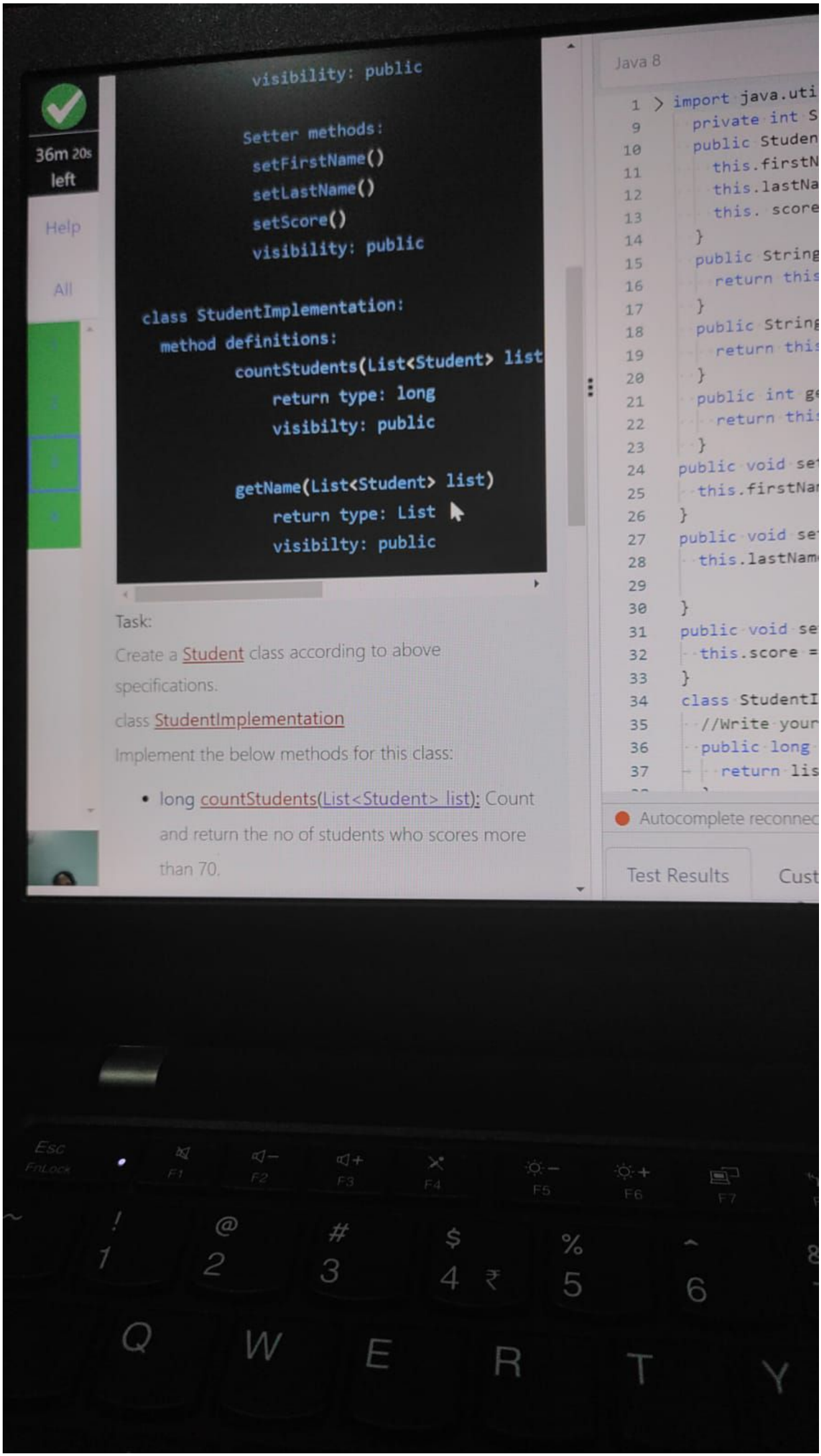
Autocomplete reconnecting...

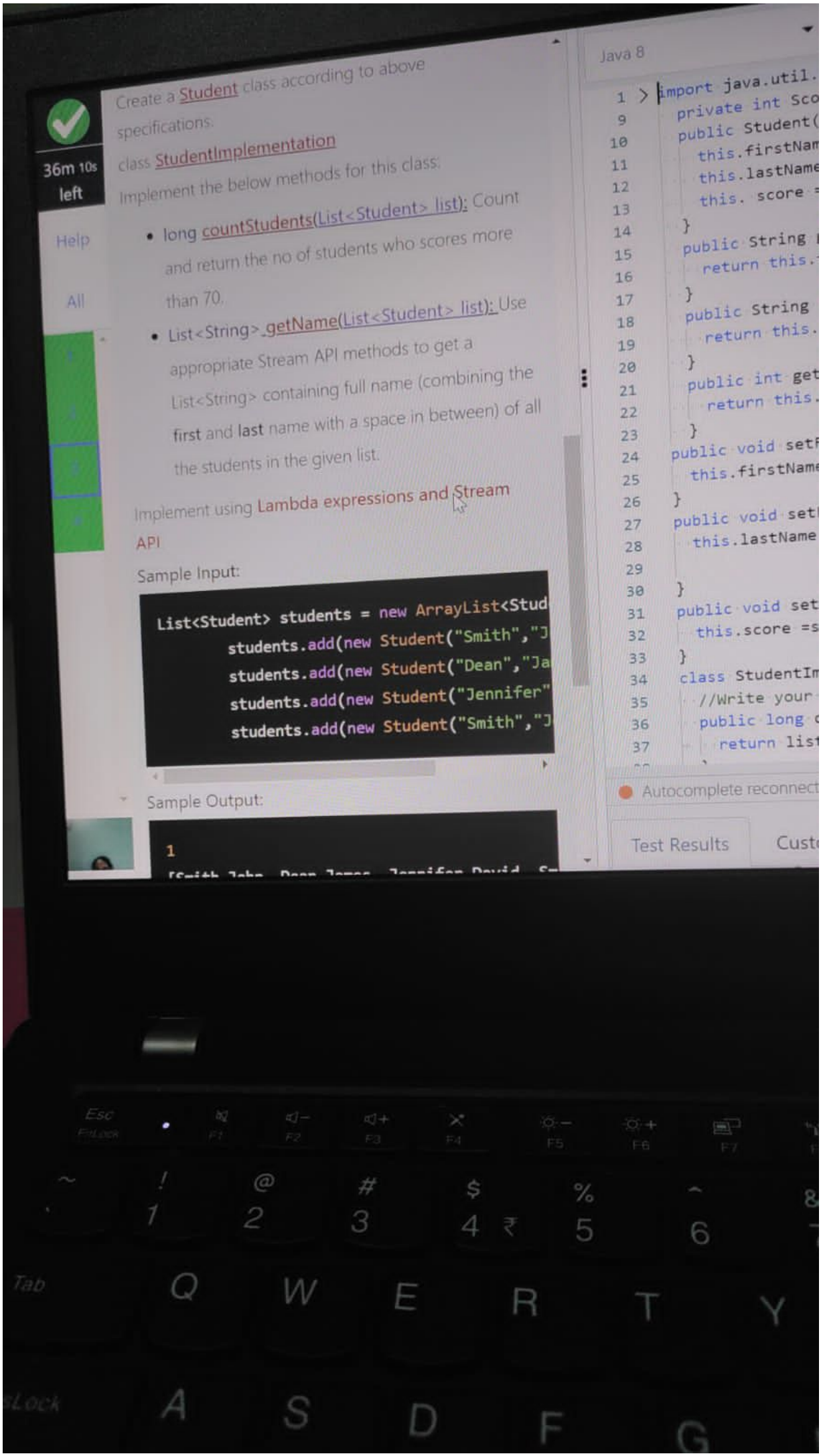
Ln 2

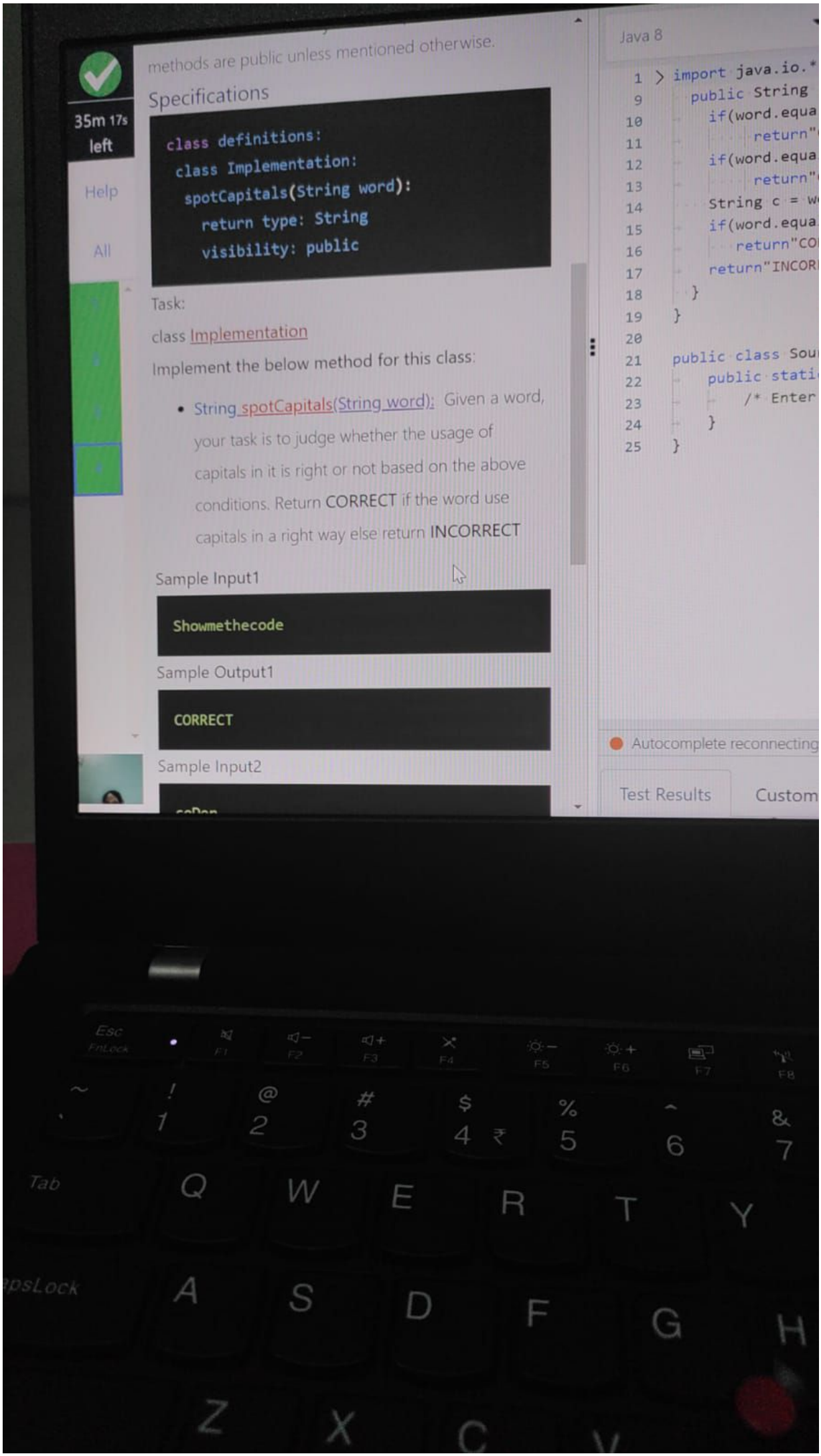
Test Results

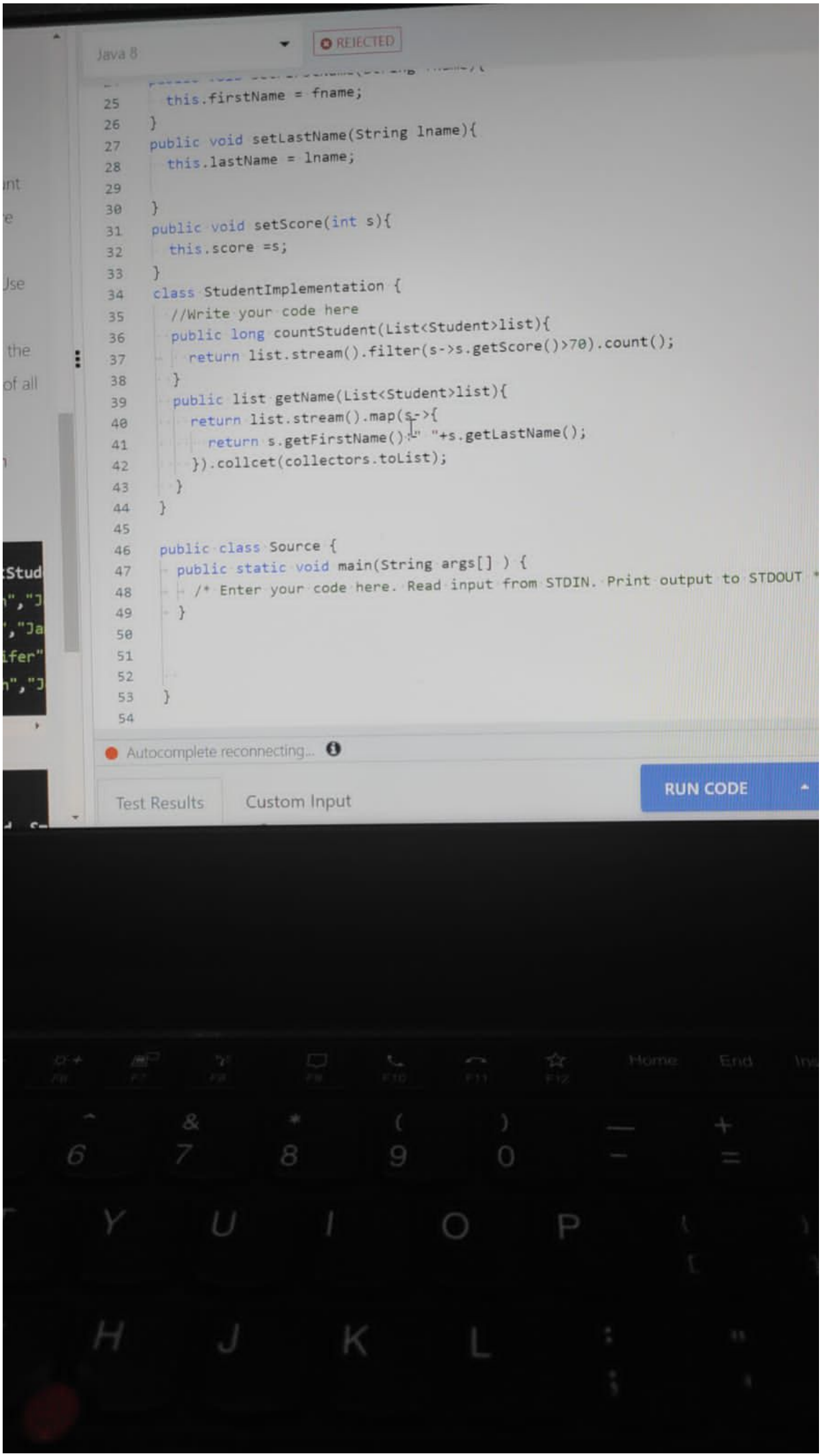
Custom Input

RUN CODE


visibility: public
Setter methods:
setFirstName()
setLastName()
setScore()
visibility: public
class StudentImplementation:
method definitions:
countStudents(List<Student> list
return type: long
visibilty: public
getName(List<Student> list)
return type: List
visibilty: public
36m 20s left
Help
All
Java 8
Task:
Create a Student class according to above specifications.
class StudentImplementation
Implement the below methods for this class:
long countStudents(List<Student> list): Count and return the no of students who scores more than 70.
Test Results


36m 10s left
Help
All
Create a Student class according to above specifications.
class StudentImplementation
Implement the below methods for this class:
long countStudents(List<Student> list): Count and return the no of students who scores more than 70.
List<String> getName(List<Student> list): Use appropriate Stream API methods to get a List<String> containing full name (combining the first and last name with a space in between) of all the students in the given list.
Implement using Lambda expressions and Stream API
Sample Input:
List<Student> students = new ArrayList<Stud
students.add(new Student("Smith","J
students.add(new Student("Dean","Ja
students.add(new Student("Jennifer"
students.add(new Student("Smith","J
Sample Output:
1
Java 8
Autocomplete reconnect
Test Results


35m 17s left
Help
All
methods are public unless mentioned otherwise.
Specifications
class definitions:
class Implementation:
spotCapitals(String word):
return type: String
visibility: public
Task:
class Implementation
Implement the below method for this class:
String spotCapitals(String word): Given a word, your task is to judge whether the usage of capitals in it is right or not based on the above conditions. Return CORRECT if the word use capitals in a right way else return INCORRECT
Sample Input1
Showmethecode
Sample Output1
CORRECT
Sample Input2
Java 8
1 > import java.io.*
9 public String
10 if(word.equa
11 return"
12 if(word.equa
13 return"
14 String c = w
15 if(word.equa
16 return"CO
17 return"INCOR
18 }
19 }
20
21 public class Sou
22 public stati
23 /* Enter
24 }
25 }
Autocomplete reconnecting
Test Results
Custom


Java 8
REJECTED
this.firstName = fname;
}
public void setLastName(String lname){
this.lastName = lname;
}
public void setScore(int s){
this.score =s;
}
class StudentImplementation {
//Write your code here
public long countStudent(List<Student>list){
return list.stream().filter(s->s.getScore()>70).count();
}
public list getName(List<Student>list){
return list.stream().map(s->{
return s.getFirstName()+" "+s.getLastName();
}).collcet(collectors.toList);
}
}
public class Source {
public static void main(String args[]) {
/* Enter your code here. Read input from STDIN. Print output to STDOUT
}
}
Autocomplete reconnecting...
Test Results
Custom Input
RUN CODE

35m 25s left

Help

All

1

2

3

4

4. Splash Uppercase

Coding

Description

Garry has recently learned about strings in her programming classes and the correct usage of capitals in grammar class.

Garry's English teacher defines the usage of capitals in a word to be right when one of the following cases holds:

- All letters in this word are capital, like "RUSSIA".
- All letters in this word are not capital, like "doselect".
- Only the first letter in this word is capital, like "Code".

Garry wants to code this and he needs your help.

Your task here is to implement a **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner.

Consider **default visibility** of classes, data fields and methods are public unless mentioned otherwise.

Specifications

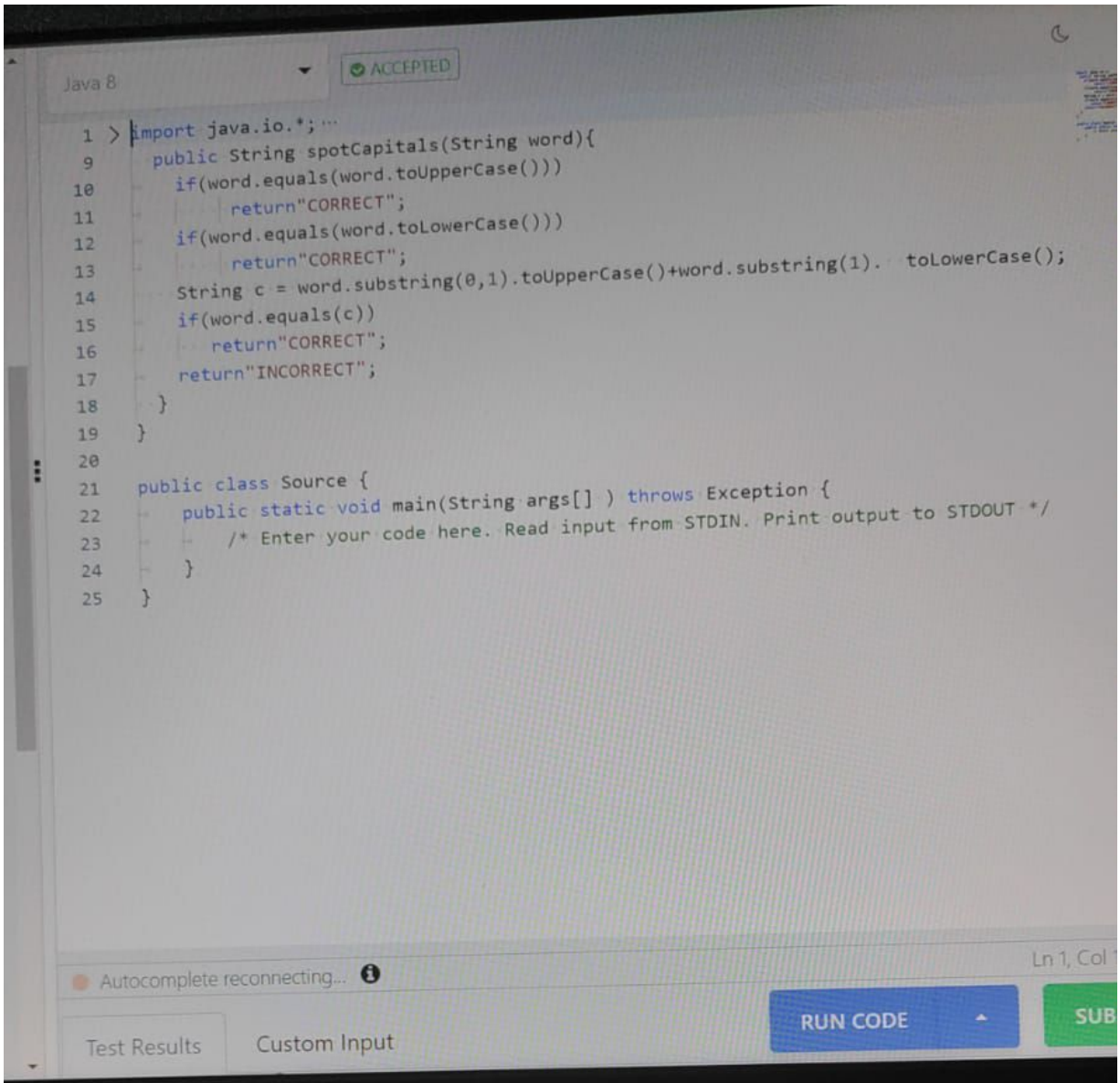
```
class definitions:
 class Implementation:
  spotCapitals(String word):
```

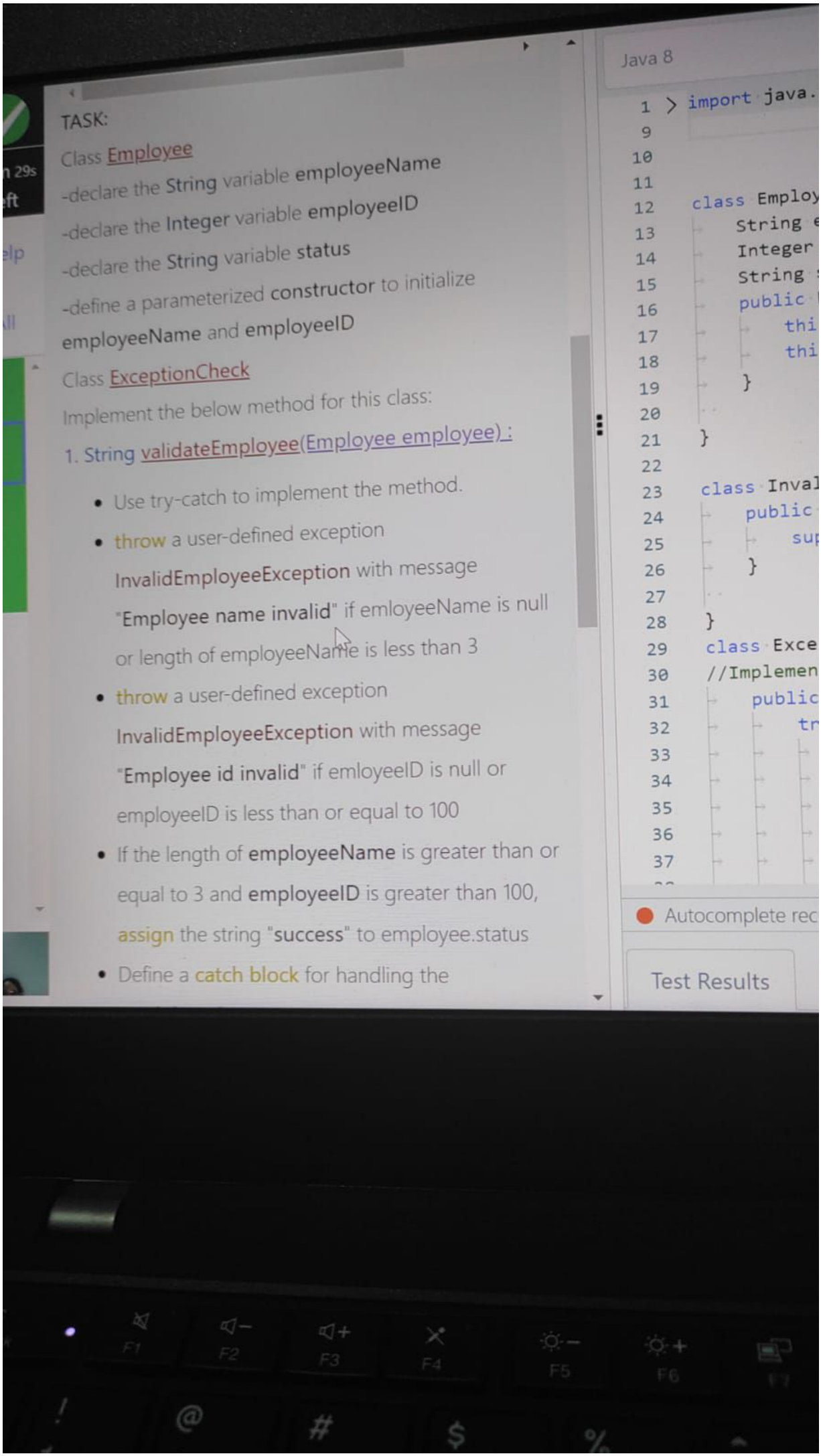
Java 8

```
1  import java.io.*;
9      public String spc
10       if(word.equals(
11           return"COR
12       if(word.equals(
13           return"COR
14       String c = word
15       if(word.equals(
16         return"CORRE
17       return"INCORRE
18     }
19   }
20
21   public class Sourc
22       public static
23           /* Enter y
24       }
25   }
```

Autocomplete reconnecting...

Test Results | Custom


Java 8
ACCEPTED
1 import java.io.*;...
9 public String spotCapitals(String word){
10 if(word.equals(word.toUpperCase()))
11 return"CORRECT";
12 if(word.equals(word.toLowerCase()))
13 return"CORRECT";
14 String c = word.substring(0,1).toUpperCase()+word.substring(1). toLowerCase();
15 if(word.equals(c))
16 return"CORRECT";
17 return"INCORRECT";
18 }
19 }
20
21 public class Source {
22 public static void main(String args[]) throws Exception {
23 /* Enter your code here. Read input from STDIN. Print output to STDOUT */
24 }
25 }
Autocomplete reconnecting...
Ln 1, Col
Test Results
Custom Input
RUN CODE
SUB



TASK:

Class Employee

-declare the String variable employeeName

-declare the Integer variable employeeID

-declare the String variable status

-define a parameterized constructor to initialize employeeName and employeeID

Class ExceptionCheck

Implement the below method for this class:

1. String validateEmployee(Employee employee) :

- Use try-catch to implement the method.
- throw a user-defined exception InvalidEmployeeException with message "Employee name invalid" if emloyeeName is null or length of employeeName is less than 3
- throw a user-defined exception InvalidEmployeeException with message "Employee id invalid" if emloyeeID is null or employeeID is less than or equal to 100
- If the length of employeeName is greater than or equal to 3 and employeeID is greater than 100, assign the string "success" to employee.status
- Define a catch block for handling the

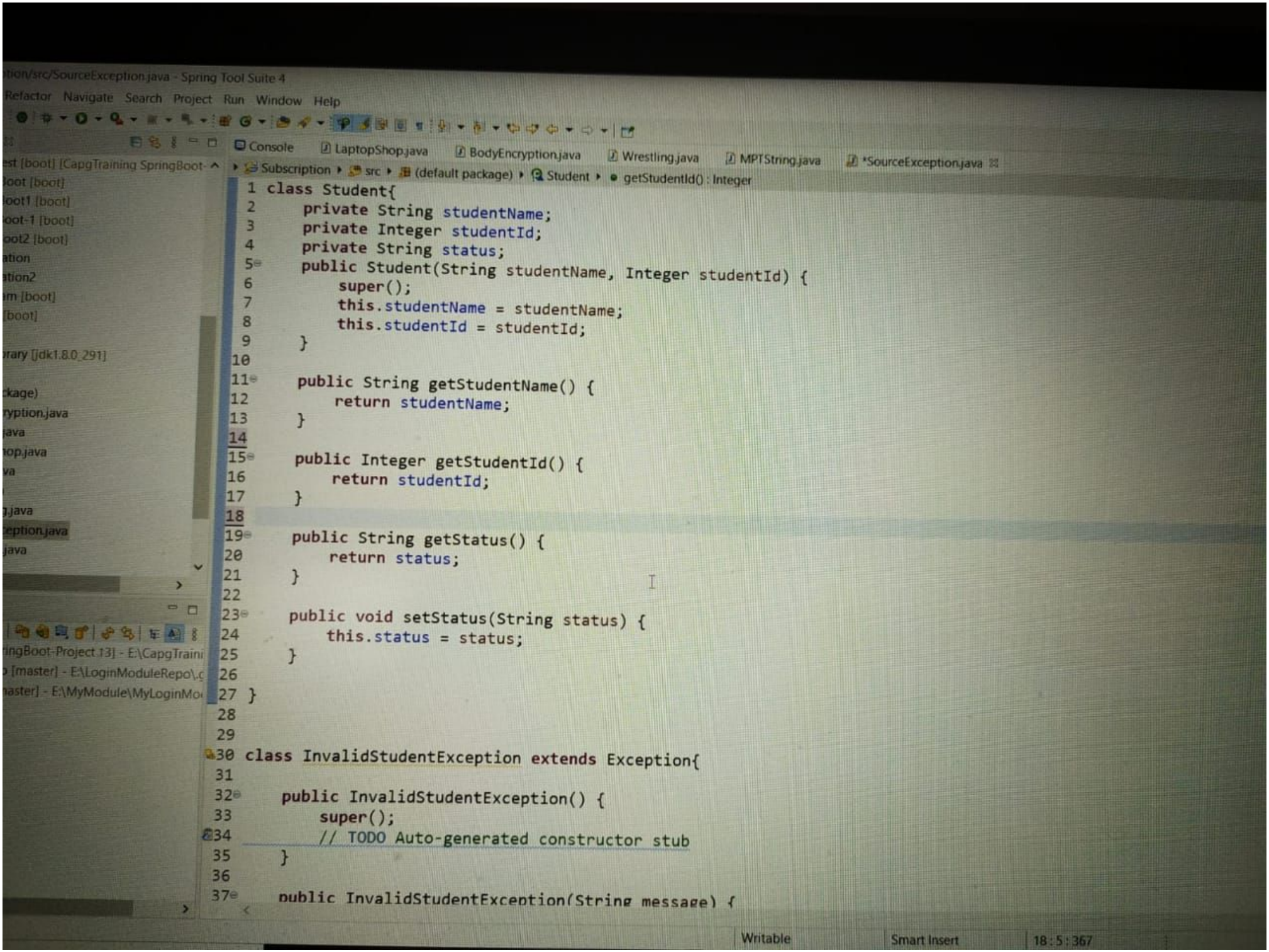
Java 8

```
1 > import java.
9
10
11
12  class Employ
13      String e
14      Integer
15      String
16      public
17          thi
18          thi
19      }
20
21  }
22
23  class Inval
24      public
25          sup
26      }
27
28  }
29  class Exce
30  //Implemen
31      public
32          tr
33
34
35
36
37
```

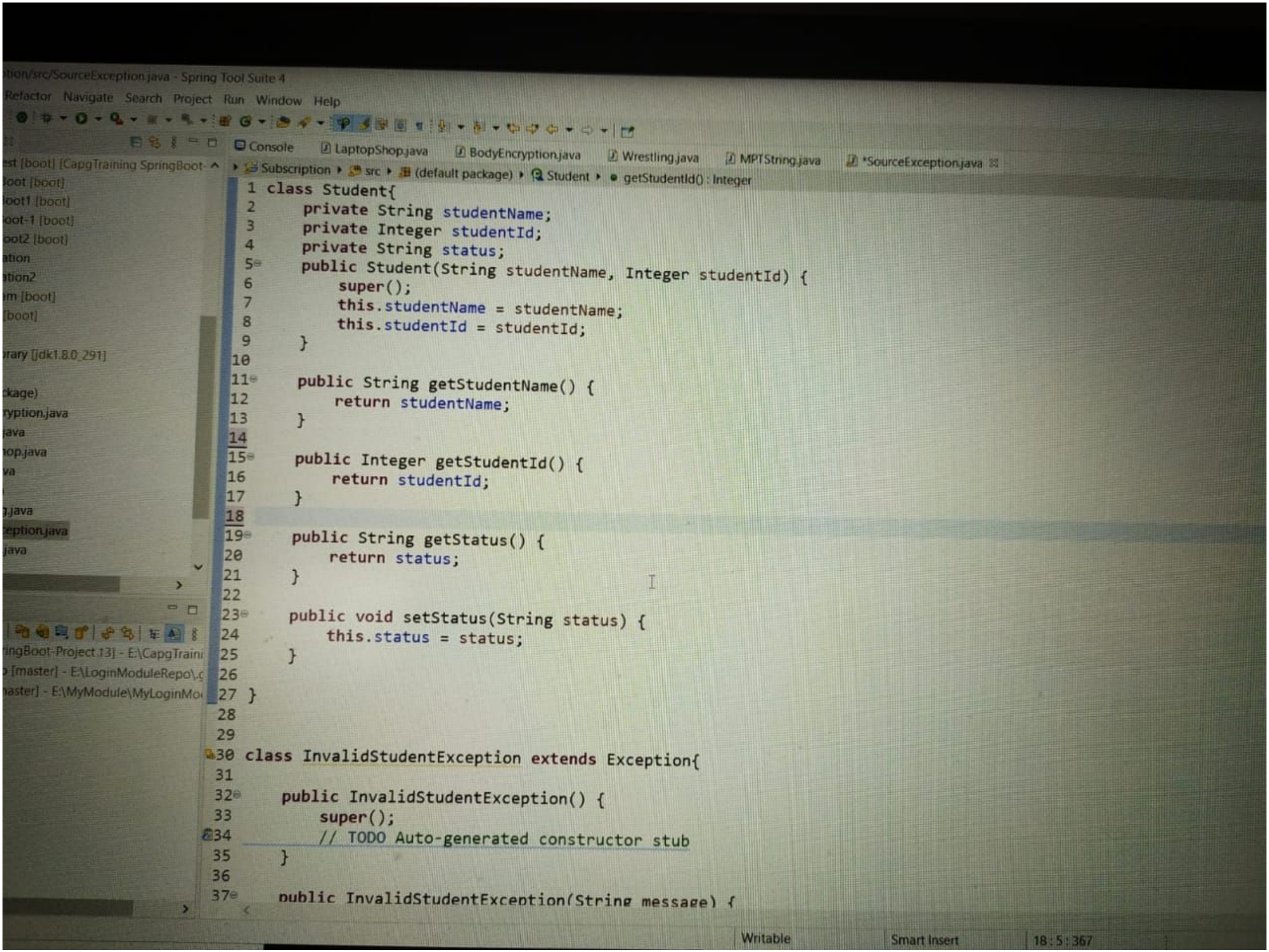
Autocomplete rec

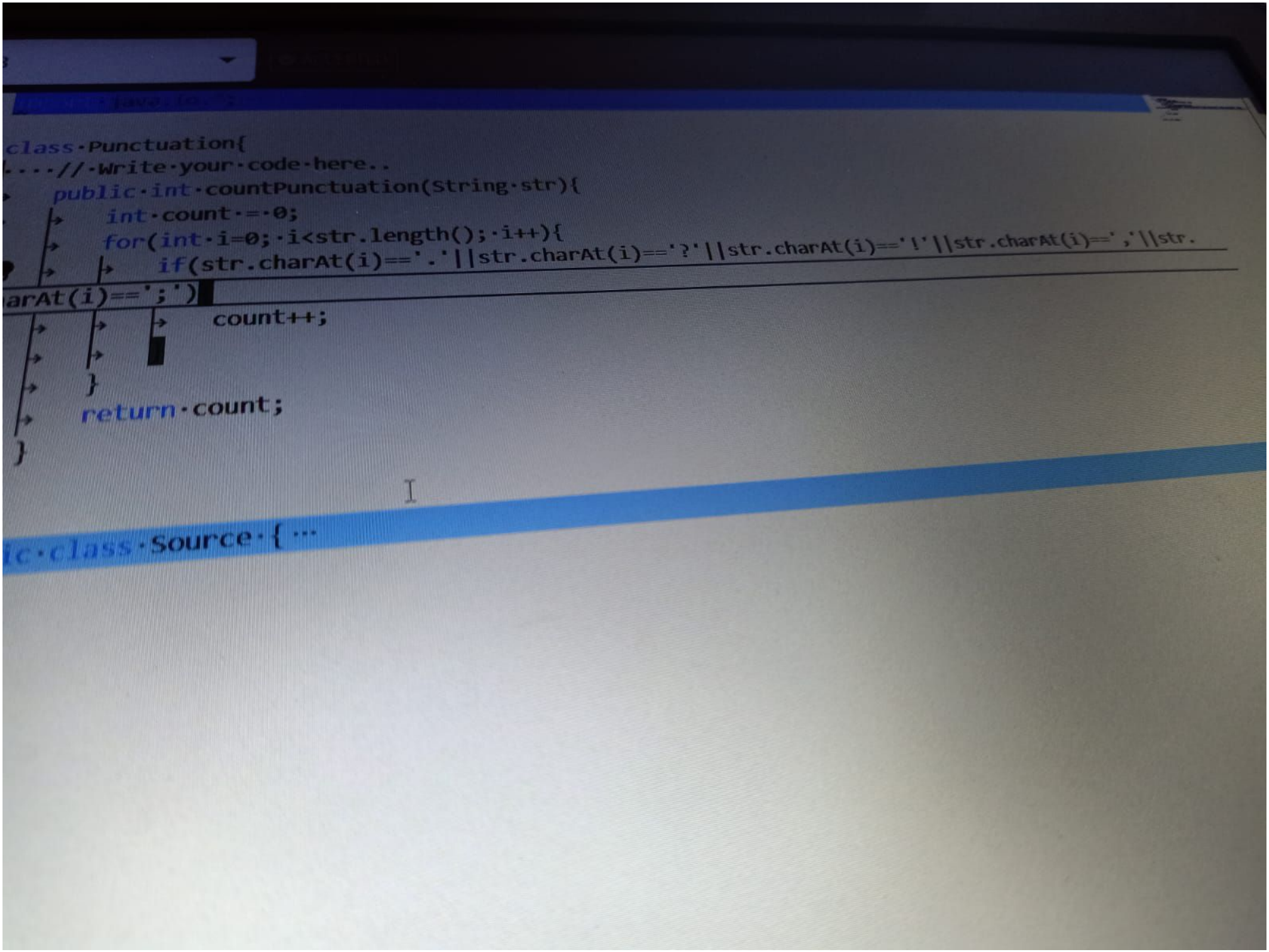
Test Results

```
class BMICalculator{

    float getWeight(String str){
        String[] arr = str.split("&");
        String []wt=arr[0].split("/");
        float res=0;
        int a=Integer.parseInt(wt[0]);
        float b=Integer.parseInt(wt[1])/(float)Math.pow(10,wt[1].length());
        res=a+b;
        return res;

    }

    float getHeight(String str){
        String[] arr = str.split("&");
        String []wt=arr[1].split("/");
        float res=0;
        int a=Integer.parseInt(wt[0]);
        float b=Integer.parseInt(wt[1])/(float)Math.pow(10,wt[1].length());
        res=a+b;
        return res;
    }
```


tion/src/SourceException.java - Spring Tool Suite 4
Refactor Navigate Search Project Run Window Help
Console
LaptopShop.java
BodyEncryption.java
Wrestling.java
MPTString.java
*SourceException.java
Subscription ▸ src ▸ (default package) ▸ Student ▸ getStudentId() : Integer
class Student{
private String studentName;
private Integer studentId;
private String status;
public Student(String studentName, Integer studentId) {
super();
this.studentName = studentName;
this.studentId = studentId;
}
public String getStudentName() {
return studentName;
}
public Integer getStudentId() {
return studentId;
}
public String getStatus() {
return status;
}
public void setStatus(String status) {
this.status = status;
}
}
class InvalidStudentException extends Exception{
public InvalidStudentException() {
super();
// TODO Auto-generated constructor stub
}
public InvalidStudentException(String message) {
Writable
Smart Insert
18 : 5 : 367

```
import java.text.*;
import java.math.*;
import java.util.regex.*;

class Product{
    //Write your code here..
    ArrayList<String> productList;
    Product()
    {
        productList=new ArrayList<String>();
    }
    void addProduct(String pName)
    {
        productList.add(pName);

    }
    void removeProduct(String pName)
    {
        for(int i=0;i<productList.size();i++)
        {
            if(productList.get(i).equals(pName))
            {
                productList.remove(i);
                break;
            }
        }

    }
    int uniqueProduct()
    {
        HashSet<String> newSet=new HashSet<String>();
        newSet.addAll(productList);
    }
```


tion/src/SourceException.java - Spring Tool Suite 4
Refactor Navigate Search Project Run Window Help
Console
LaptopShop.java
BodyEncryption.java
Wrestling.java
MPTString.java
*SourceException.java
Subscription ▸ src ▸ (default package) ▸ Student ▸ getStudentId() : Integer
class Student{
private String studentName;
private Integer studentId;
private String status;
public Student(String studentName, Integer studentId) {
super();
this.studentName = studentName;
this.studentId = studentId;
}
public String getStudentName() {
return studentName;
}
public Integer getStudentId() {
return studentId;
}
public String getStatus() {
return status;
}
public void setStatus(String status) {
this.status = status;
}
}
class InvalidStudentException extends Exception{
public InvalidStudentException() {
super();
// TODO Auto-generated constructor stub
}
public InvalidStudentException(String message) {
Writable
Smart Insert
18 : 5 : 367

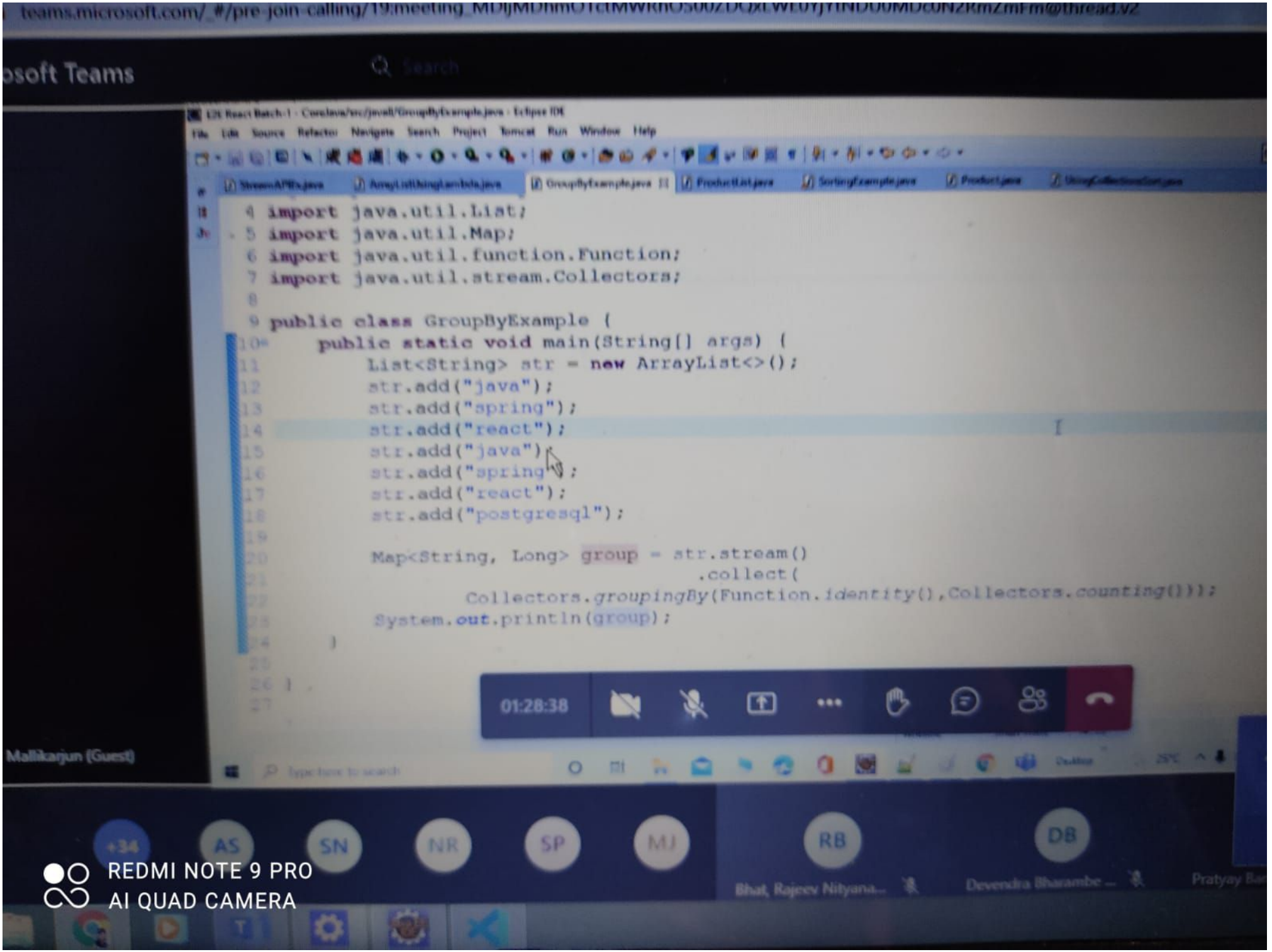

class Punctuation{
//Write your code here..
public int countPunctuation(String str){
int count = 0;
for(int i=0; i<str.length(); i++){
if(str.charAt(i)=='.'||str.charAt(i)=='?'||str.charAt(i)=='!'||str.charAt(i)==','||str.
arAt(i)==';')
count++;
}
return count;
}
lic class Source { ...

```
class Implementation {
    //Write your code here..
    public String getIndex(ArrayList<String> seriesList,int index){
        return seriesList.get(index);
    }
    public ArrayList<String> addAfterSeries(ArrayList<String> seriesList,String p,String q){
        ArrayList<String> newList=new ArrayList<>();
        seriesList.forEach((l)->{
            if(l.equals(p)){
                newList.add(l);
                newList.add(q);
            }
            else{
                newList.add(l);
            }
        });
        return newList;
    }
}

class Source{
    public static void main(String[] args){
        ArrayList<String> seriesList=new ArrayList<>();
        seriesList.add("Breaking Bad");
        seriesList.add("Game of Thrones");
        seriesList.add("Friends");
```

```
    }
}

class Source{
    public static void main(String[] args){
        ArrayList<String> seriesList=new ArrayList<>();
        seriesList.add("Breaking Bad");
        seriesList.add("Game of Thrones");
        seriesList.add("Friends");
        seriesList.add("Prison break");
        Scanner sc=new Scanner(System.in);
        Implementation imp=new Implementation();
        String p=sc.nextLine();
        String q=sc.nextLine();
        int n=sc.nextInt();
        System.out.println(imp.getIndex(seriesList,n));
        System.out.println(imp.addAfterSeries(seriesList,p,q));
    }
}
```

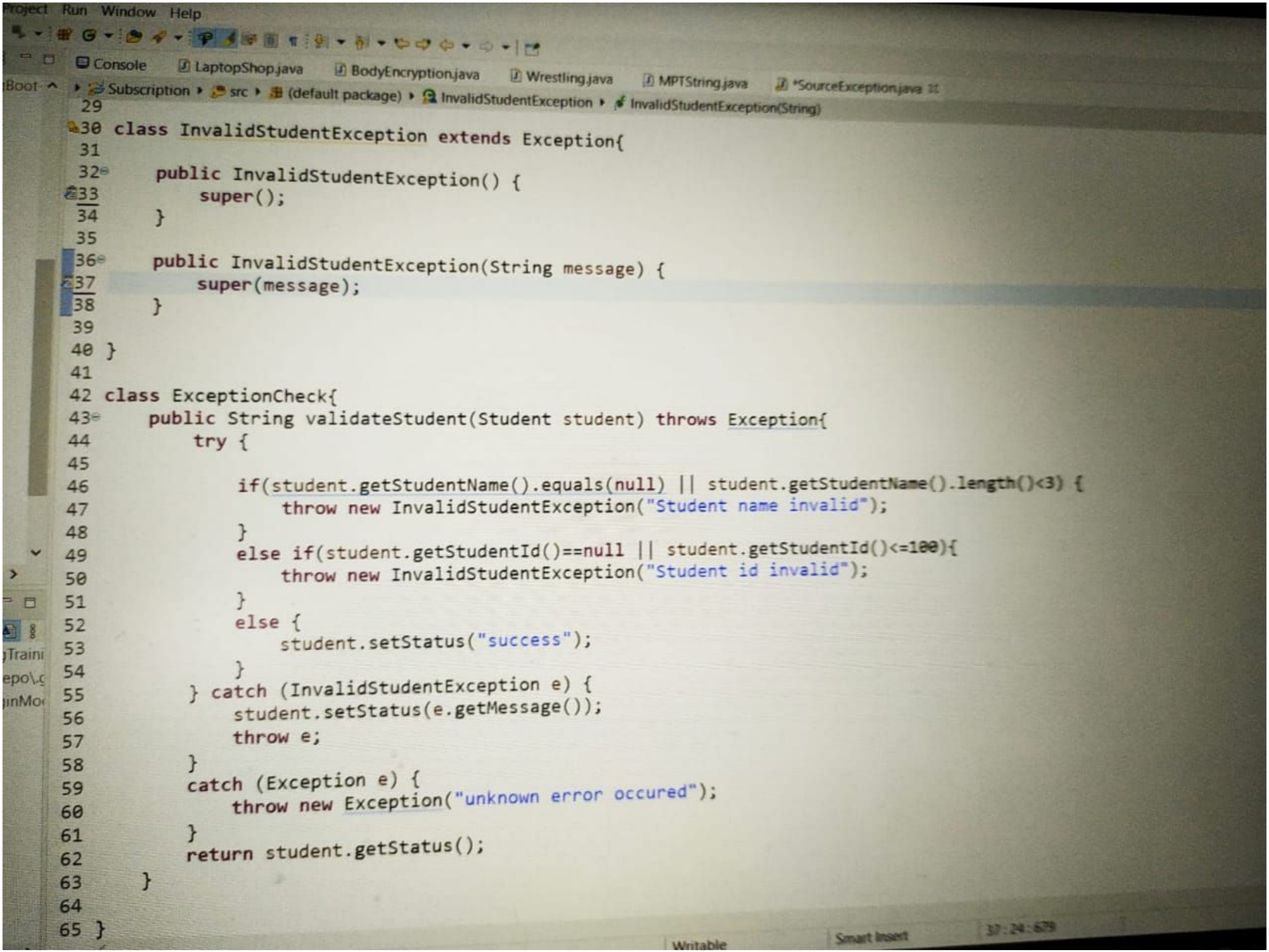
Autocomplete ready

```
29
30      public ArrayList<String> changeOccurence(ArrayList<String> a, String m, String n){
31          for (int i = 0; i<a.size(); ++i) {
32              if (a.get(i).equals(m)) a.set(i, n);
33          }
34          return a;
35      }
36
37      public String listIndex(ArrayList<String>  list) {
38          return list.get(0);
39      }
40
41      public ArrayList<String> listAfter(ArrayList<String> a, String m, String n){
42          for (int i = 0; i<a.size(); ++i){
43              if (a.get(i).equals(m)) a.add(i+1, n);
44          }
45          return a;
46      }
```


teams.microsoft.com/_#/pre-join-calling/19:meeting_MDjMDhmOTctMWRhOS00ZDQtZWE0YjYtNDU0MDc0N2RmZmFm@thread.v2
osoft Teams
Search
L2E React Batch-1 - Corejava/src/javall/GroupByExample.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
StreamAPIEx.java
ArrayListUsingLambda.java
GroupByExample.java
ProductList.java
SortingExample.java
Product.java
4 import java.util.List;
5 import java.util.Map;
6 import java.util.function.Function;
7 import java.util.stream.Collectors;
8
9 public class GroupByExample {
10 public static void main(String[] args) {
11 List<String> str = new ArrayList<>();
12 str.add("java");
13 str.add("spring");
14 str.add("react");
15 str.add("java");
16 str.add("spring");
17 str.add("react");
18 str.add("postgresql");
19
20 Map<String, Long> group = str.stream()
21 .collect(
22 Collectors.groupingBy(Function.identity(),Collectors.counting()));
23 System.out.println(group);
24 }
25
26 }
27
01:28:38
Mallikarjun (Guest)
+34
AS
SN
NR
SP
MJ
RB
DB
Bhat, Rajeev Nityana...
Devendra Bharambe ...
Pratyay Bu
REDMI NOTE 9 PRO
AI QUAD CAMERA

valuation logs

```
Source.java:48: error: cannot find symbol
 int m=list.stream().collect(Collectors.groupingBy(k
                                                   I
  symbol:   variable comparator
  location: class WrestleImplementation
```


Project Run Window Help
Console
LaptopShop.java
BodyEncryption.java
Wrestling.java
MPTString.java
*SourceException.java
Subscription ▸ src ▸ (default package) ▸ InvalidStudentException ▸ InvalidStudentException(String)
29
30 class InvalidStudentException extends Exception{
31
32 public InvalidStudentException() {
33 super();
34 }
35
36 public InvalidStudentException(String message) {
37 super(message);
38 }
39
40 }
41
42 class ExceptionCheck{
43 public String validateStudent(Student student) throws Exception{
44 try {
45
46 if(student.getStudentName().equals(null) || student.getStudentName().length()<3) {
47 throw new InvalidStudentException("Student name invalid");
48 }
49 else if(student.getStudentId()==null || student.getStudentId()<=100){
50 throw new InvalidStudentException("Student id invalid");
51 }
52 else {
53 student.setStatus("success");
54 }
55 } catch (InvalidStudentException e) {
56 student.setStatus(e.getMessage());
57 throw e;
58 }
59 catch (Exception e) {
60 throw new Exception("unknown error occured");
61 }
62 return student.getStatus();
63 }
64
65 }
Writable
Smart Insert

Java 8 ACCEPTED

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class Shipping{
    // Write your code here ..
     String  sourcePlace;
     String destinationPlace;
     int netWeight;
     int totalWeight;
     public Shipping(String  sourcePlace,String destinationPlace,int netWeight,int totalWeight)
     {
         this.sourcePlace=sourcePlace;
         this.destinationPlace=destinationPlace;
         this.netWeight=netWeight;
         this.totalWeight=totalWeight;
     }
}

class Implementation{
    // Write your code here ..

    public String validator(Shipping details) throws Exception
    {
        if(details.sourcePlace.equals(details.destinationPlace))
        {
            throw new SamePlaceException("source and destination cannot be same");
        }
   else if(details.netWeight > details.totalWeight)
    {
    throw new WeightException("net weight cannot be greater than total weight");
    }
    else
{
    return "shipping details are valid";
```

Autocomplete reconnecting.

Test Results Custom Input

RUN CODE

```
        return (float) 0.0;
        else
        return (float)-1.0;
    }
            }
            catch(Exception e){
                return (float)0.0;  }

            return totalBill;
    }
    }


    class SamePlaceException extends Exception{
        // Write your code here ..
        public SamePlaceException(String message)
        {
            super(message);
        }
    }

    class  WeightException extends Exception{
        // Write your code here ..
        public WeightException(String message)
        {
    super(message);
        }
    }

    public class Source {
        public static void main(String args[] ) throws Exception {
            /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        }
    }
```

Autocomplete reconnecting...

Test Results | Custom Input

RUN CODE

F7 F8 F9 F10 F11 F12 Prt Sc Insert Delete

Java 8 ACCEPTED

```
37      return "shipping details are valid";
38      }
39
40      }
41      public float totalBill(Shipping details)
42      {
43          float totalBill = 0.0f;
44          try
45          {
46
47  String result = validator(details);
48      if(result.equalsIgnoreCase("shipping details are valid"))
49                  totalBill = details.totalWeight*5;
50  else{
51      if(details.sourcePlace.equalsIgnoreCase(details.destinationPlace)||
52      details.netWeight> details.  totalWeight)
53      return (float) 0.0;
54      else
55      return (float)-1.0;
56  }
57          }
58          catch(Exception e){
59              return (float)0.0;  }
60
61          return totalBill;
62  }
63  }
64
65
66
67  class SamePlaceException extends Exception{
68      // Write your code here ..
69      public SamePlaceException(String message)
70      {
71          super(message);
72      }
73  }
```

Autocomplete reconnecting...

Test Results | Custom Input

RUN CODE

ASUS VivoBook

F8 F9 F10 F11 F12 Prt Sc Insert Delete

```java
import java.text.SimpleDateFormat;
import java.util.Date;

class Batch{

    static boolean lengthCheck(String code){
        if(code.length()==12)
        return true;
        return false;
    }

    static boolean batchNumberCheck(String code){
        String batch=code.substring(0,4);
        String checkone=batch.substring(0,1);
        String checktwo=batch.substring(1,2);
        String checkthree=batch.substring(3,4);
        char x=checkone.charAt(0);
        char y=checktwo.charAt(0);
        char z=checkthree.charAt(0);
        if((Character.isUpperCase(x)&&Character.isUpperCase(y))&&Character.isUpperCase(z)){
            try{
                int intValue=Integer.parseInt(batch.substring(2,3));
                return true;
            }catch(NumberFormatException e){
                System.out.println("Input String cannot parsed to Integer,");
            }
        }
        return false;
    }

    static boolean yearCheck(String code){
        String batch=code.substring(4,8);
        int year = Integer.parseInt(batch);
        if(year>=2015 && year<=2020)
        {return true;
        }else {
            return false;
```

Autocomplete reconnecting...

Ln 1, Col 1 Java 8

Test Results | Custom Input

RUN CODE | SUBMIT

```
    static boolean yearCheck(String code){
        String batch=code.substring(4,8);
        int year = Integer.parseInt(batch);
        if(year>=2015 && year<=2020)
        {return true;
        }else {
            return false;
        }
    }

    static boolean monthCheck(String code){
        String batch=code.substring(8,10);
        int month=Integer.parseInt(batch);
        if(month>=1 && month<=12){
            return true;
        }else{
            return false;
        }
    }

    static boolean dayCheck(String code){
        String batch=code.substring(10,12);
        int day=Integer.parseInt(batch);
        if(day>=1 && day<=31){
            return true;
        }else{
            return false;
        }
    }

}

class Source{
    public static void main(String[] args) {
        //Don't touch this or else your program might not run...
    }
}
```