```java
package progusingjava;

//DO NOT MODIFY THE CODE PROVIDED TO YOU

class BusBooking extends Booking
{
    public static String[] startDestinationArr={"FLO-CAL","CAL-GEO","CAL-WAS"};//Florida, California,Georgia,Washington
    public static int[] busFareArr= {200,400,600};
    private String busType;
    private double totalBusFare;
    private String day;

    public BusBooking(String busType,Passenger passenger,String source,String destination,boolean returnJourney,String day)
    {
        super(passenger,source,destination,returnJourney);
        this.busType=busType;
        this.totalBusFare=0;
        this.day=day;
    }

    @Override
    public String toString() {
        return "BusBooking("+"busType: " + this.busType + ", passenger: "
                + this.getPassenger() + ", source: " + this.getSource() + ", destination: " + this.getDestination()
                + ", returnJourney: " + this.getReturnJourney() + ", day: " + this.day+")";
    }

    public double getTotalBusFare()
    {
        return this.totalBusFare;
    }

    public int identifyChargedPercentage(String busType)
    {
        int percentageCharge=0;
        if(busType.equalsIgnoreCase("ORDINARY"))
```

```
    int percentageCharge=0;
    if(busType.equalsIgnoreCase("ORDINARY"))
    {
        percentageCharge=5;
    }
    else if(busType.equalsIgnoreCase("SEMISLEEPER"))
    {
        percentageCharge=10;
    }
    else if(busType.equalsIgnoreCase("SLEEPER"))
    {
        percentageCharge=20;
    }
    else
    {
        percentageCharge=-1;
    }
    return percentageCharge;
}

//To Trainee
public int findBaseFare()
{
    int baseFare=0;

    //Implement your logic here
    boolean flag= false;
    int b=0;
    String onwardJourney= getSource().substring(0,3)+"-"+ getDestination().substring(0,3);
    String returnJourney= getDestination().substring(0,3)+"-"+ getSource().substring(0,3);
    for(int i=0; i<startDestinationArr.length; i++){
        if(onwardJourney.equalsIgnoreCase(startDestinationArr[i])|| returnJourney.equalsIgnoreCase(startDestinationArr[i])){
            b=busFareArr[i];
            flag= true;
        }
    }
```

```
67                  b=busFareArr[i];
68                  flag= true;
69              }
70          }
71          if(flag){
72              baseFare=b;
73          }
74          else{
75              baseFare=-1;
76          }
77          return baseFare;
78      }
79
80      //To Trainee
81      public void calculateTotalBusFare()
82      {
83          if(!getPassenger().validatePassengerDetails()){
84              this.totalBusFare=-1;
85              setPnrNumber("NA");
86          }
87          else{
88              int taxPercentage= identifyChargedPercentage(this.busType);
89              if(findBaseFare()!=-1 && taxPercentage!=-1){
90                  double finalFare= findBaseFare()+(findBaseFare()*taxPercentage/100);
91                  if(day.equalsIgnoreCase("friday")|| day.equalsIgnoreCase("saturday")||day.equalsIgnoreCase("sunday")){
92                      finalFare=finalFare+30;
93                  }
94                  if(getPassenger().getAge()>=60){
95                      finalFare-=(finalFare*50)/100;
96                  }
97                  if(getReturnJourney()){
98                      finalFare= finalFare*2+20;
99                  }
100                 else{
101                     finalFare= finalFare+10;
102                 }
```

Question | Booking.java | BusBooking.java ×

```
        int taxPercentage= identifyChargedPercentage(this.busType);
        if(findBaseFare()!=-1 && taxPercentage!=-1){
            double finalFare= findBaseFare()+(findBaseFare()*taxPercentage/100);
            if(day.equalsIgnoreCase("friday")|| day.equalsIgnoreCase("saturday")||day.equalsIgnoreCase("sunday")){
                finalFare=finalFare+30;
            }
            if(getPassenger().getAge()>=60){
                finalFare-=(finalFare*50)/100;
            }
            if(getReturnJourney()){
                finalFare= finalFare*2+20;
            }
            else{
                finalFare= finalFare+10;
            }
            totalBusFare=finalFare;
            generatePnrNumber();
        }
        else{
            totalBusFare=-1;
            setPnrNumber("NA");
        }
    }
    }

    //Implement your logic here

}
```

File Edit Selection View Go Run Terminal Exam Help

Reference Question Booking.java × BusBooking.java

```
27
28      public String getPnrNumber() {
29          return this.pnrNumber;
30      }
31
32      public void setPnrNumber(String pnrNumber) {
33          this.pnrNumber = pnrNumber;
34      }
35      public Passenger getPassenger() {
36          return this.passenger;
37      }
38      public String getSource() {
39          return this.source;
40      }
41      public String getDestination() {
42          return this.destination;
43      }
44
45      //To Trainee
46      public void generatePnrNumber()
47      {
48
49          //Implement your logic here
50          char ch= getPassenger().getName().charAt(0);
51          String pnrNo= Character.toString(ch).toUpperCase()+ ++counter;
52          setPnrNumber((pnrNo));
53
54      }
55
56      public boolean getReturnJourney() {
57          return this.returnJourney;
58      }
59  }
60
61
62
```

Reference | Question | Booking.java | Passenger.java × | BusBooking.java

```
34
35      public String getPhoneNo() {
36          return this.phoneNo;
37      }
38
39      public String getEmailId() {
40          return this.emailId;
41      }
42
43      //To Trainee
44      public Boolean validatePassengerDetails()
45      {
46          boolean flag= false;
47          //Implement your logic here
48          if(getAge()>12 && getName().length()>= 3 && getPhoneNo().charAt(0)=='9'){
49              for(String str: domainArr){
50                  if(emailId.contains(str)){
51                      flag= true;
52                      break;
53                  }
54              }
55              return flag;
56          }
57          else {
58              return flag;
59          }
60
61
62
63
64          //Change the return statement accordingly
65
66      }
67  }
68
69
```

Edit Selection View Go Run Terminal Exam Help

rence Question Booking.java Passenger.java Tester.java × BusBooking.java

```java
package progusingjava;
public class Tester
{
    Run | Debug
    public static void main(String args[])
    {
        Passenger passengerObj=new Passenger("Peter",26,"9865768743","abc123@gmail.com");
        BusBooking busBookingObj=new BusBooking("SEMISLEEPER",passengerObj,"florida","california",true,"Saturday");
        busBookingObj.calculateTotalBusFare();
        System.out.println("PNR Number:"+busBookingObj.getPnrNumber());
        System.out.println("Total Bus Fare:"+busBookingObj.getTotalBusFare());
    }
}
```

va BusBooking.java × Solution.java Verification ×

;

CODE PROVIDED TO YOU

ends Booking

ring[] startDestinationArr={"
nt[] busFareArr= {200,400,600}
usType;
totalBusFare;
day;

ng(String busType,Passenger pa

nger,source,destination,return
pe=busType;
BusFare=0;
ay;

toString() {
usBooking("+"busType: " + this.b
this.getPassenger() + ", source
", returnJourney: " + this.getR

getTotalBusFare()

is.totalBusFare;

dentifyChargedPercentage(String b

tentageCharge=0;
ype.equalsIgnoreCase("ORDINARY"))

Structural - Non Assessed

Test case results against structural verification.
Functional test cases are relevant only if all structural test cases pass.

100%

View Results (13/13)

Functional - Non Assessed

Test case results against logical verification.
These results can be used to debug your code.
These are not assessed.

100%

View Results (14/14)

Functional - Assessed

Test case results against logical verification.
These are assessed.
Test results are not shown.

100%

(31/31)

00:21:49 Ln 4, Col 1 LF UTF-8 Spaces: 4 Java

Reference | Question | Queue.java | Solution.java ×

```
1  package dsausingjava;
2
3  //DO NOT MODIFY THE CODE PROVIDED TO YOU
4
   Verify
5  public class Solution {
6      public Stack conversionInt(Stack inStrStack, Queue inIntQueue){
7          Stack outStrStack=new Stack(inStrStack.getMaxSize());
8
9          //Implement your logic here
10         Stack stck= new Stack(inStrStack.getMaxSize());
11         int x= 0;
12         int y=0;
13         int z=0;
14         while(!inStrStack.isEmpty()){
15             x= inIntQueue.dequeue();
16             y= inIntQueue.dequeue();
17             z= x+y;
18             String str1= inStrStack.pop();
19             String str2= String.valueOf(z);
20             String str3= String.valueOf((z)*2);
21             char chr= str2.charAt(str2.length()-1);
22             String str4= String.valueOf(chr);
23             if(str1.contains(str2)){
24                 outStrStack.push(str1);
25             }
26             else if(str1.contains(str4)){
27                 outStrStack.push(str1+str3);
28             }
29             else{
30                 stck.push(str1);
31             }
32         }
33         while(!stck.isEmpty()){
34             outStrStack.push(stck.pop());
35         }
```

NjA3NmFmMTYwM2VmNDMzZTY5OTk3NDY2IHNhc3dhdC5wYW5pZ3JhaGkwMQ==

00:22:35 Ln 1, Col 1 LF

```
18                  String str1= inStrStack.pop();
19                  String str2= String.valueOf(z);
20                  String str3= String.valueOf((z)*2);
21                  char chr= str2.charAt(str2.length()-1);
22                  String str4= String.valueOf(chr);
23                  if(str1.contains(str2)){
24                      outStrStack.push(str1);
25                  }
26                  else if(str1.contains(str4)){
27                      outStrStack.push(str1+str3);
28                  }
29                  else{
30                      stck.push(str1);
31                  }
32              }
33              while(!stck.isEmpty()){
34                  outStrStack.push(stck.pop());
35              }
36
37          return outStrStack;
38      }
39
40  }
```

Reference | Question | Queue.java | Solution.java | Tester.java ×

```java
package dsausingjava;

public class Tester {
    Run | Debug
    public static void main(String[] args){
        Solution obj1 =new Solution();

        Stack inStrStack= new Stack(4);
        inStrStack.push("123");
        inStrStack.push("kil8");
        inStrStack.push("tr5");
        inStrStack.push("999");

        Queue inIntQueue=new Queue(8);
        inIntQueue.enqueue(3);
        inIntQueue.enqueue(6);
        inIntQueue.enqueue(17);
        inIntQueue.enqueue(8);
        inIntQueue.enqueue(31);
        inIntQueue.enqueue(6);
        inIntQueue.enqueue(8);
        inIntQueue.enqueue(9);

        Stack outStrStack=obj1.conversionInt(inStrStack,inIntQueue);
        outStrStack.display();
    }
}
```

```
gjava;

 THE CODE PROVIDED TO YOU


lution {
k conversionInt(Stack inStrStack,
utStrStack=new Stack(inStrStack.ge

ement your logic here
stck= new Stack(inStrStack.getMaxSi
 0;
0;
0;
!inStrStack.isEmpty()){
 inIntQueue.dequeue();
 inIntQueue.dequeue();
 x+y;
ring str1= inStrStack.pop();
ring str2= String.valueOf(z);
ring str3= String.valueOf((z)*2);
ar chr= str2.charAt(str2.length()-1
ring str4= String.valueOf(chr);
(str1.contains(str2)){
   outStrStack.push(str1);

lse if(str1.contains(str4)){
   outStrStack.push(str1+str3);

lse{
   stck.push(str1);


le(!stck.isEmpty()){
 outStrStack.push(stck.pop());
```

Structural - Non Assessed

Test case results against structural verification.
Functional test cases are relevant only if all structural test cases pass.

100

View Results (3/3)

Functional - Non Assessed

Test case results against logical verification.
These results can be used to debug your code.
These are not assessed.

100%

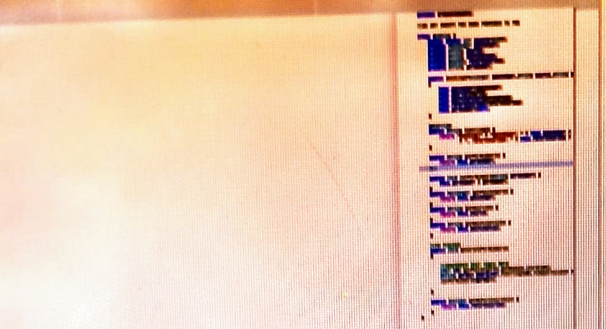
View Results (3/3)

Functional - Assessed

Test case results against logical verification.
These are assessed.
Test results are not shown.

100%

(8/8)



```
String source,String d

ey;

: " + this.passenger +
+ this.returnJourney +

nrNumber) {

{
```

Submission successfull!

Submit again, if changes are made.

View Details (48/48)