

```
Reference Question Customer.java x DigitalItemOrder.java MusicOrder.java Tester.java
13         this.emailId=emailId;
14         this.age=age;
15     }
16
17     //To Trainee
18     public Boolean validateCustomerDetails(){
19         //Implement your logic here
20         if(this.custName.length()>=4 && this.age>18 && this.emailId.endsWith(".com")){
21             return true;
22         }else
23             I
24         //Change the return statement accordingly
25         return false;
26     }
27
28     public String getCustName(){
29         return this.custName;
30     }
31
32     public String getEmailId(){
33         return this.emailId;
34     }
35 }
```

NjA2YWNlODQwM2VmNDMzZTY5ZjcwYzUwIGFiaGlzaGVrLmRob25kZ2U= 00:59:36 Ln 26, Col 6 LF UTF-8 Spaces: 4 Java

↓ 6.1 Kbps xctored!
↑ 2.0 Kbps

```

36     }
37
38     public void setOrderPrice(double orderPrice){
39         this.orderPrice=orderPrice;
40     }
41
42     //To Trainee
43     public void generateOrderId(){
44
45         //Implement your logic here
46         char ch = this.getItemName().charAt(0);
47         String j= Character.toString(ch);
48         orderId= j + (++counter);
49     }
50
51     @Override
52     public String toString() {
53         return "itemName: " + this.itemName + "customer: " + this.customer + ", orderPrice:
54         + this.orderPrice + ", orderId: " + this.orderId;
55     }
56     public abstract void calculateOrderPrice();
57

```

NjA2YWNlODQwM2VmNDMzZTY5ZjcwYzUwIGFiaGlzaGVrLmRob25kZ2U= 00:59:08 Ln 48, Col 32 LF UTF-8

7.2 Kbps
3.5 Kbps

	Reference	Question	Customer.java	DigitalItemOrder.java	MusicOrder.java x	Trainer.java
201	24	}				
	25					
Trainee	26	//To Trainee				
	27	public Integer validateBitRate(){				
sausingjava	28	//Implement your logic here				
ueue.java	29	if (this.bitRate==192){				
olution.java	30	return 0;				
ack.java	31	}				
ter.java	32	else if (this.bitRate==128){				
usingjava	33	return 1;				
omer.java	34	}				
allItemOr...	35	else if (this.bitRate==256){				
Order.j...	36	return 2;				
java	37	}				
	38	else{				
	39	return -1;				
	40	}				
	41					
NCIES	42	//Change the return statement accordingly				
5fad4c	43					
	44	}				
	45					

NjA2YWNlODQwM2VmNDMzZTY5ZjcwYzUwIGFiaGlzaGVrLmRob25kZ2U= 00:58:54 Ln 72, Col 32 LF

6.9 Kbps
2.7 Kbps

```
Selection View Go Run Terminal Exam Help
Customer.java DigitalItemOrder.java MusicOrder.java x Tester.java
50 //Implement your logic here
51 boolean format = false;
52 for (int a=0 ; a<musicFormatAvailable.length ; a++){
53     if( musicFormatAvailable[a].equalsIgnoreCase(musicFormat)){
54         format = true;
55     }
56 }
57 if(this.getCustomer().validateCustomerDetails()&&format){
58     int bitRateIndex=this.validateBitRate();
59     if(bitRateIndex!= -1 && durationInSec>0){
60         double baseCost = bitRateCost[(int)validateBitRate()];
61         int division = durationInSec/60;
62         baseCost= baseCost+3*(division);
63         baseCost = baseCost +(5*(baseCost))/100;
64         this.setOrderPrice(baseCost);
65         this.generateOrderId();
66     }else{
67         this.setOrderPrice(-1.0);
68         this.setOrderId("NA");
69     }
70 }else{
71     this.setOrderPrice(-1.0);

```

NjA2YWNIODQwM2VmNDMzZTY5ZjcwYzUwIGFiaGlzaGVrLmRob25kZ2U= 00:58:41 Ln 72, Col 32 LF UTF-8

6.3 Kbps xctored!
8.0 Kbps

Reference	Question	Customer.java	DigitalItemOrder.java	MusicOrder.java	Tester.java
63			baseCost = baseCost + (5*(baseCost))/100;		
64			this.setOrderPrice(baseCost);		
65			this.generateOrderId();		
66			}else{		
67			this.setOrderPrice(-1.0);		
68			this.setOrderId("NA");		
69			}		
70			}else{		
71			this.setOrderPrice(-1.0);		
72			this.setOrderId("NA");		
73			}		
74			}		
75			}		

NjA2YWNlODQwM2VmNDMzZTY5ZjcwYzUwIGFiaGlzaGVrLmRob25kZ2U= 00:58:35 Ln 72, Col 32 LF UTF-

6.3 Kbps
3.3 Kbps

```
Run View Go Run Terminal Exam Help
Reference Question Customer.java DigitalItemOrder.java MusicOrder.java Solution.java x Test.java
6 public Stack stackRearrangement(Stack inIntStack, Queue inIntQueue) {
7     Stack outIntStack = new Stack(inIntStack.getMaxSize()*2);
8
9     //Implement your logic here
10    Stack ab= new Stack(inIntStack.getMaxSize());
11    Stack bc= new Stack(3);
12    while (!inIntStack.isEmpty()){
13        ab.push(inIntStack.pop());
14    }
15    while (!inIntQueue.isEmpty()){
16        bc.push(inIntQueue.dequeue());
17    }
18    for (int i=0; i<3+ab.getMaxSize(); i++){
19        if (i==0 || i == (3+ab.getMaxSize())/2 || i==2+ab.getMaxSize())
20            outIntStack.push(bc.pop());
21        else
22            outIntStack.push(ab.pop());
23    }
24    return outIntStack;
25 }
26
```

NjA2YWNlODQwM2VmNDMzZTY5ZjcwYzUwIGFiaGlzaGVrLmRob25kZ2U= 00:58:19 Ln 1, Col 1 LF UTF-8 Spaces: 4

5.7 Kbps stored! 1.5 Kbps