

CODE

INPUT

OUTPUT

Java 8 ▾

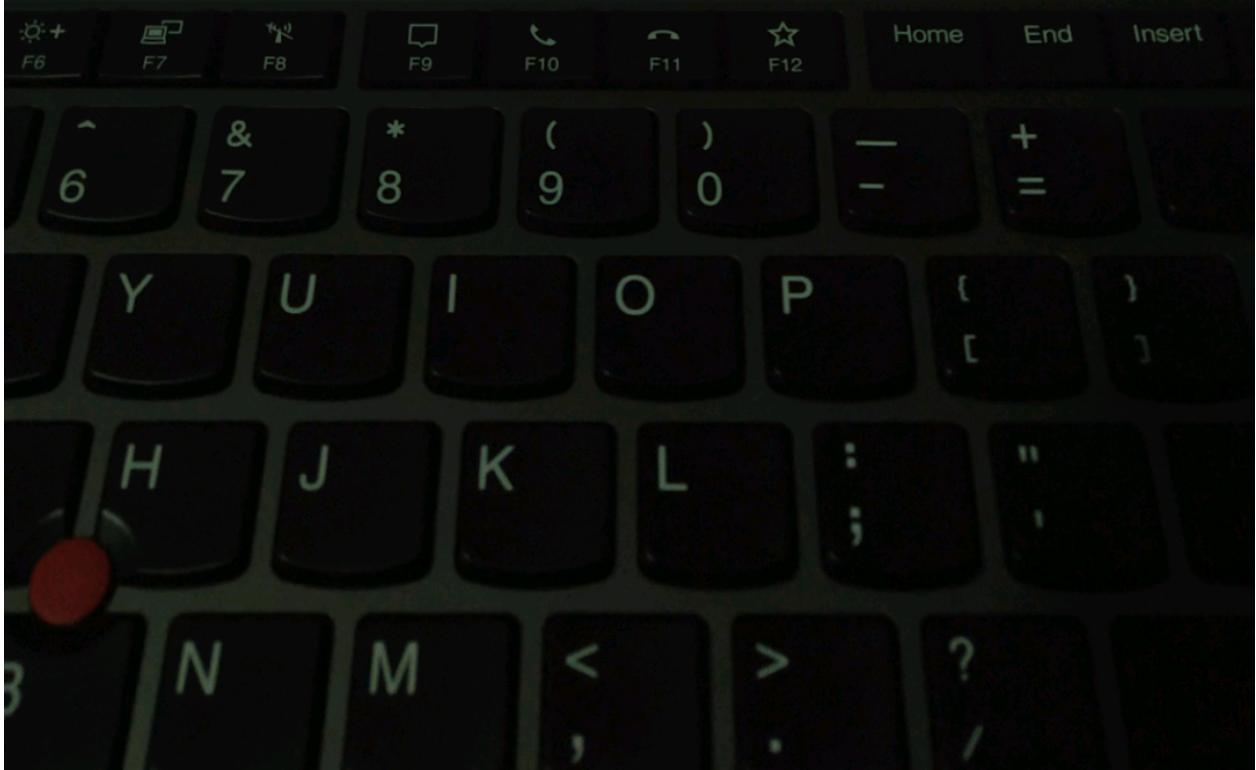
RUN CODE

⋮

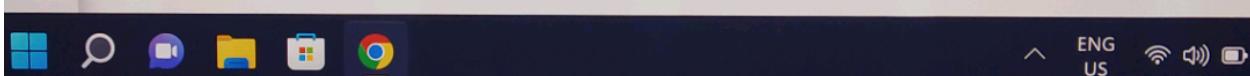
```
1 import java.util.*;
2 class CustomerNotFoundException extends RuntimeException
3 {
4
5 }
6 class Customer{
7     private int userId;
8     private String emailId;
9     private String password;
10    private String firstName;
11    private String lastName;
12    private String city;
13    private String gender;
14    private long phoneNumber;
15    private Address address;
16
17    Customer(){}
18    public Customer(int userId, String emailId, String password, String firstName, String l
19        this.userId=userId;
20        this.emailId=emailId;
21        this.password=password;
22        this.firstName=firstName;
23        this.lastName=lastName;
24        this.city=city;
25        this.gender=gender;
```



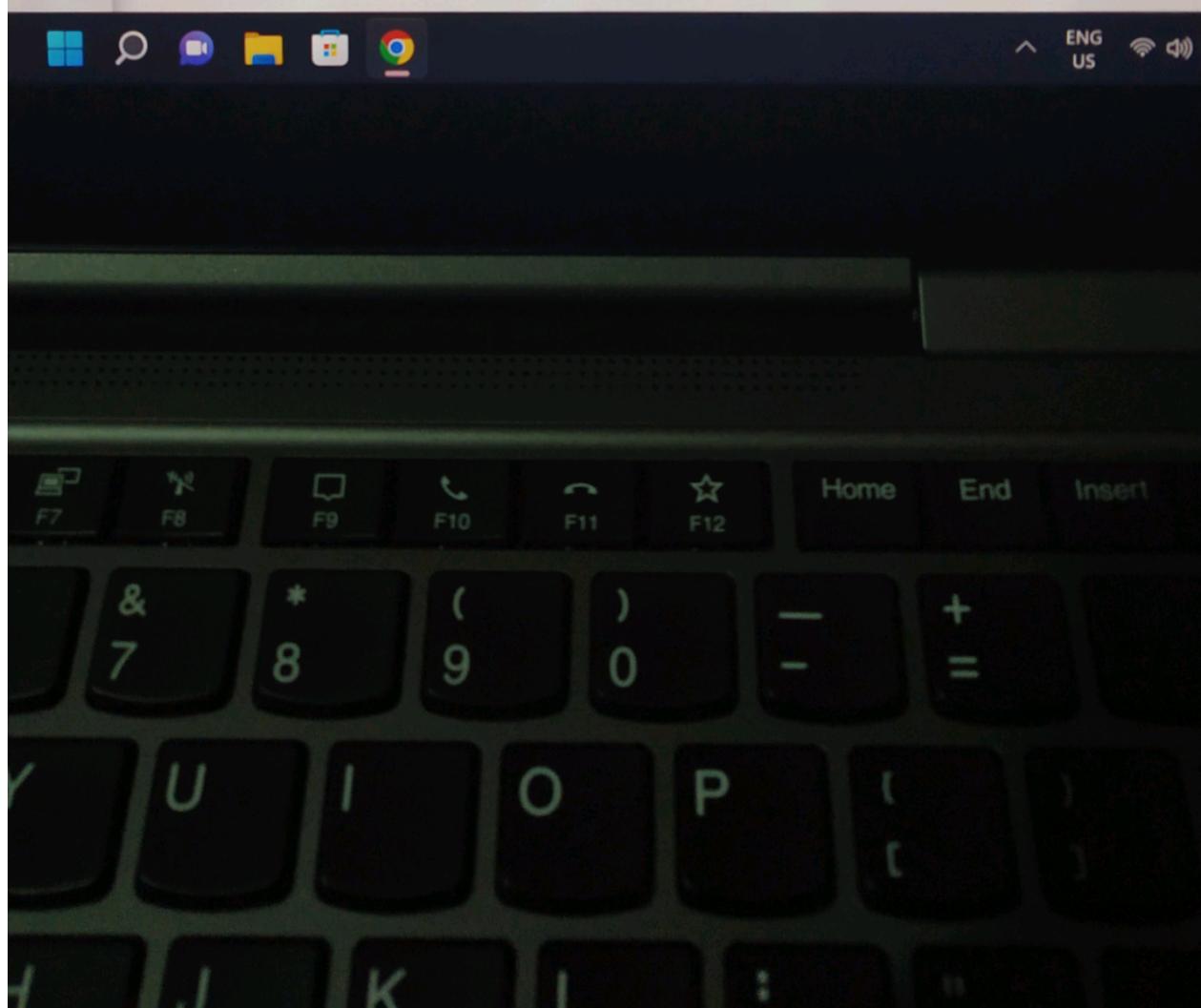
ENG
US



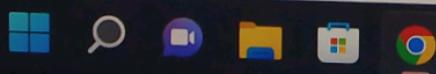
```
1
2 : RuntimeException
3
4
5
6 I
7
8
9
10
11
12
13
14
15
16
17
18 •emailId, String password, String firstName, String lastName, String city, String gender, long p
19
20
21
22
23
24
25
26
27
28
```



```
1  
2  
3  
4  
5  
6 I  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18 -me, String lastName, String city, String gender, long phoneNumber, Address address){  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28
```

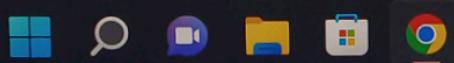


```
17     Customer(){}
18  public Customer(int userId, String emailId, String password, String
19      this.userId=userId;
20      this.emailId=emailId;
21      this.password=password;
22      this.firstName=firstName;
23      this.lastName=lastName;
24      this.city=city;
25      this.gender=gender;
26      this.phoneNumber=phoneNumber;
27      this.address=address;
28  }
29  public int getUserId(){
30      return userId;
31  }
32  public void setId(int userId){
33      this.userId=userId;
34  }
35  public String getEmailId(){
36      return emailId;
37  }
38  public void setEmailId(String emailId){
39      this.emailId=emailId;
40  }
41  public String getPassword(){
42      return password;
43  }
44  public void setPassword(String password){
45      this.password=password;
46 }
```



```
40      }
41  public String getPassword(){
42      return password;
43  }
44  public void setPassword(String password){
45      this.password=password;
46  }
47  public String getFirstName(){
48      return firstName;
49  }
50  public void setFirstName(String firstName){
51      this.firstName=firstName;
52  }
53  public String getLastName(){
54      return lastName;
55  }
56  public void setLastName(String lastName){
57      this.lastName=lastName;
58  }
59  public String getCity(){
60      return city;
61  }
62  public void setCity(String city){
63      this.city=city;
64  }
65  public String getGender(){
66      return gender;
67  }
68  public void setGender(String gender){
69      this.gender=gender;

```



F6 F7 F8 F9 F10 F11 F12 Home

& * ()

```
practicesession/186717/cidb/full/view&type=assessment&amode=new&attemptNum=2&assessment=57571&com... 🔍 ⭐
```

```
60         return city;
61     }
62     public void setCity(String city){
63         this.city=city;
64     }
65     public String getGender(){
66         return gender;
67     }
68     public void setGender(String gender){
69         this.gender=gender;
70     }
71     public long getPhoneNumber(){
72         return phoneNumber;
73     }
74     public void setPhoneNumber(long phoneNumber){
75         this.phoneNumber=phoneNumber;
76     }
77     public Address getAddress(){
78         return address;
79     }
80     public void setAddress(Address address){
81         this.address=address;
82     }
83     @Override
84     public String toString(){
85         return "Customer [userId="+userId+", emailId="+emailId+, password="+password+",
86     }
87 }
88 class Address{
89     private String city;
```

The image shows a laptop screen displaying Java code in a code editor. The code defines a `Customer` class with various getters and setters for `city`, `gender`, `phoneNumber`, and `address`. It also contains a nested `Address` class. The code editor has a dark theme with syntax highlighting. Below the screen, a portion of a black keyboard is visible, showing keys like F6-F12, Home, End, Insert, and a numeric keypad.

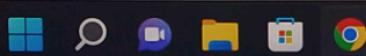
```
79     }
80     public void setAddress(Address address){
81         this.address=address;
82     }
83     @Override
84     public String toString(){
85         return "Customer [userId="+userId+", emailId="+emailId+, password="+password+",
86     }
87 }
88 class Address{
89     private String city;
90     private String state;
91     private int zip;
92     private String country;
93
94     Address(){}
95     public Address(String city,String state,int zip,String country){
96         this.city=city;
97         this.state=state;
98         this.zip=zip;
99         this.country=country;
100    }
101    public String getCity(){
102        return city;
103    }
104    public void setCity(String city){
105        this.city=city;
106    }
107    public String getState(){
108        return state;
```



ENG
US 9:3
5/28



```
79  
80 *  
81  
82  
83  
84 -  
85     llId="+emailId+", password="+password+", firstName="+firstName+", lastName="+lastName+",  
86             I  
87  
88 -  
89  
90  
91  
92  
93  
94  
95 -zip,String country){  
96  
97  
98  
99  
100  
101 -  
102  
103  
104 -  
105  
106  
107 -  
108
```

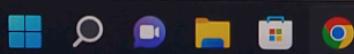


9:3
ENG
US 5/28



acticesession/186717/cidb/full/view/&type=assessment&amode=new&attemptNum=2&assessment=57571&com...    

```
79  
80 ▾  
81  
82  
83  
84 ▾  
85 :"+password+", firstName='"+firstName+", lastName='"+lastName+", city='"+city+", gender='"+ge  
86 I  
87  
88 ▾  
89  
90  
91  
92  
93  
94  
95 ▾  
96  
97  
98  
99  
100  
101 ▾  
102  
103  
104 ▾  
105  
106  
107 ▾  
108
```



^ ENG
US 9:35
Wi-Fi 5/28/2

F7 F8 F9 F10 F11 F12 Home End Insert Delete

&

7

*

8

(

9

)

0

-

-

+

=

Y

U

I

O

P

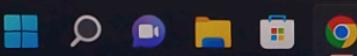
{

}

]

icesession/186717/cidb/full/view/&type=assessment&amode=new&attemptNum=2&assessment=57571&com... ↗ ☆ ⚙ D

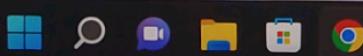
```
79
80
81
82
83
84
85 lastName)+"lastName+", city)+"city+", gender)+"gender+", phoneNumber)+"phoneNumber+", ac
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
```



9:35
ENG
US 5/28/2024



```
79  
80 ▾  
81  
82  
83  
84 ▾  
85 ', city='"+city+"', gender='"+gender+"', phoneNumber='"+phoneNumber+"', address='"+address+"']";  
86  
87  
88 ▾  
89  
90  
91  
92  
93  
94  
95 ▾  
96  
97  
98  
99  
100  
101 ▾  
102  
103  
104 ▾  
105  
106  
107 ▾  
108
```

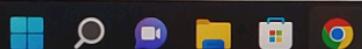


ENG
US 5/2



ADAPT-...

```
100      }
101  public String getCity(){
102      return city;
103  }
104  public void setCity(String city){
105      this.city=city;
106  }
107  public String getState(){
108      return state;
109  }
110  public void setState(String state){
111      this.state=state;
112  }
113  public int getZip(){
114      return zip;
115  }
116  public void setZip(int zip){
117      this.zip=zip;
118  }
119  public String getCountry(){
120      return country;
121  }
122  public void setCountry(String country){
123      this.country=country;
124  }
125  @Override
126  public String toString(){
127      return "Address [city='"+city+", state='"+state+", zip='"+zip+", country='"+country+
128  }
129 }
```

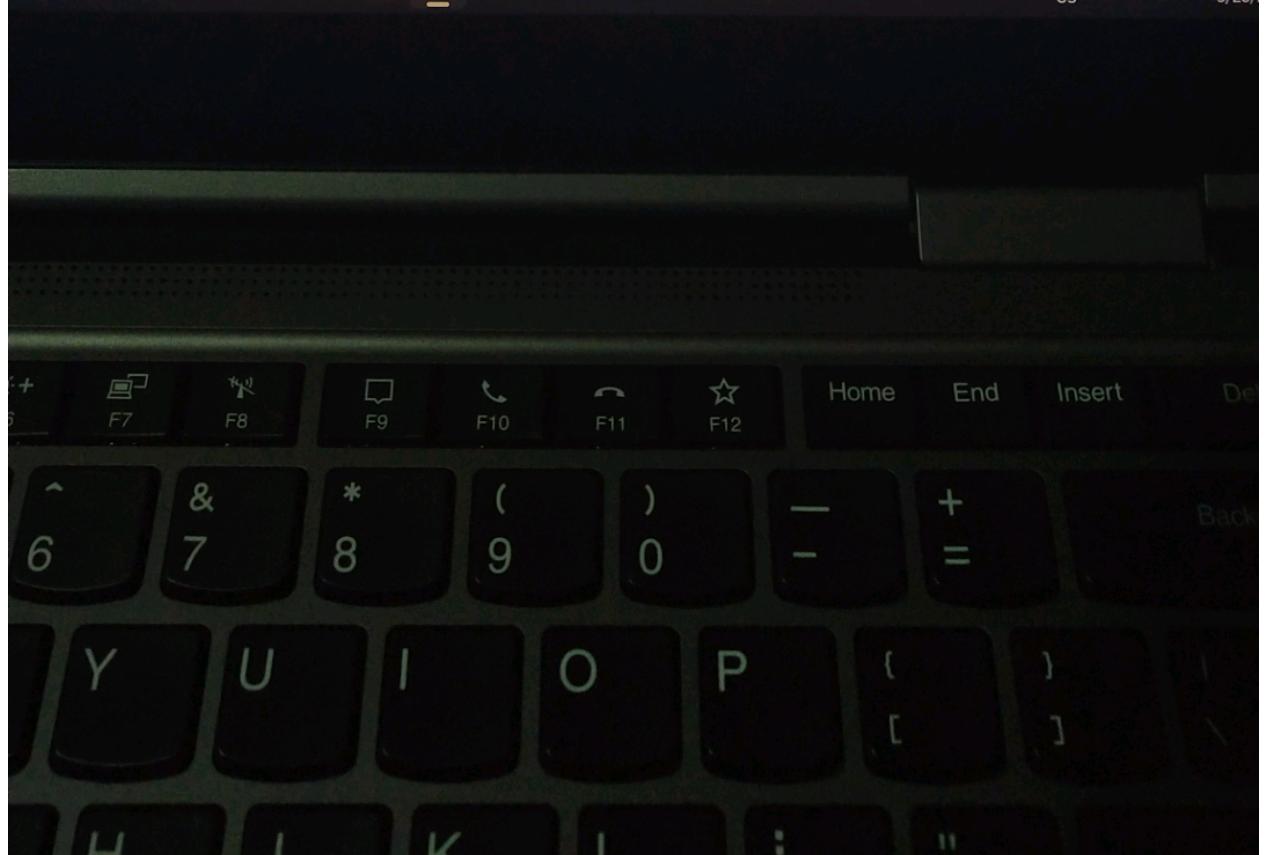


ENG
US



```
epracticesession/186717/cidb/full/view/&type=assessment&amode=new&attemptNum=2&assessment=575718&com... ↗ ⭐ ↘
```

```
122 public void setCountry(String country){  
123     this.country=country;  
124 }  
125 @Override  
126 public String toString(){  
127     return "Address [city="+city+", state="+state+", zip="+zip+", country="+country+  
128 }  
129 }  
130 interface CustomerService  
131 {  
132     void createCustomer(Customer customer);  
133     Customer updateCustomer(Customer customer);  
134     void deleteCustomer(int id);  
135     Customer searchCustomer(int id);  
136     Customer[] getCustomers();  
137 }  
138 class CustomerServiceImpl implements CustomerService  
139 {  
140     public static Customer customerArray[]=new Customer[5];  
141     public static int count=0;  
142     @Override  
143     public void createCustomer(Customer customer)  
144     {  
145         customerArray[count++]=customer;  
146     }  
147     @Override  
148     public Customer updateCustomer(Customer customer) throws CustomerNotFoundException{  
149         for(int i=0;i<5;i++)  
150         {  
151             if(customer.getUserId()==customerArray[i].getUserId()) {  
152                 customerArray[i]=customer;  
153                 count++;  
154             }  
155         }  
156     }  
157 }
```



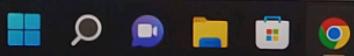
```
122 -    setCountry(String country){  
123     untry=country;  
124  
125  
126 -    ng toString(){  
127     "Address [city='"+city+"", state='"+state+"", zip='"+zip+"", country='"+country+""]";  
128  
129  
130     merService  
131 -  
132     Customer(Customer customer);  
133     dateCustomer(Customer customer);  
134     Customer(int id);  
135     archCustomer(int id);  
136     getCustomers();  
137  
138     erviceImpl implements CustomerService  
139 -  
140     ic Customer customerArray[] =new Customer[5];  
141     ic int count=0;  
142  
143     createCustomer(Customer customer)  
144 -  
145     rArray[count++]=customer;  
146  
147  
148 -     omer updateCustomer(Customer customer) throws CustomerNotFoundException{  
149     i=0;i<5;i++)  
150 -  
151 -     customer.getUserId()==customerArray[i].getUserId()){
```



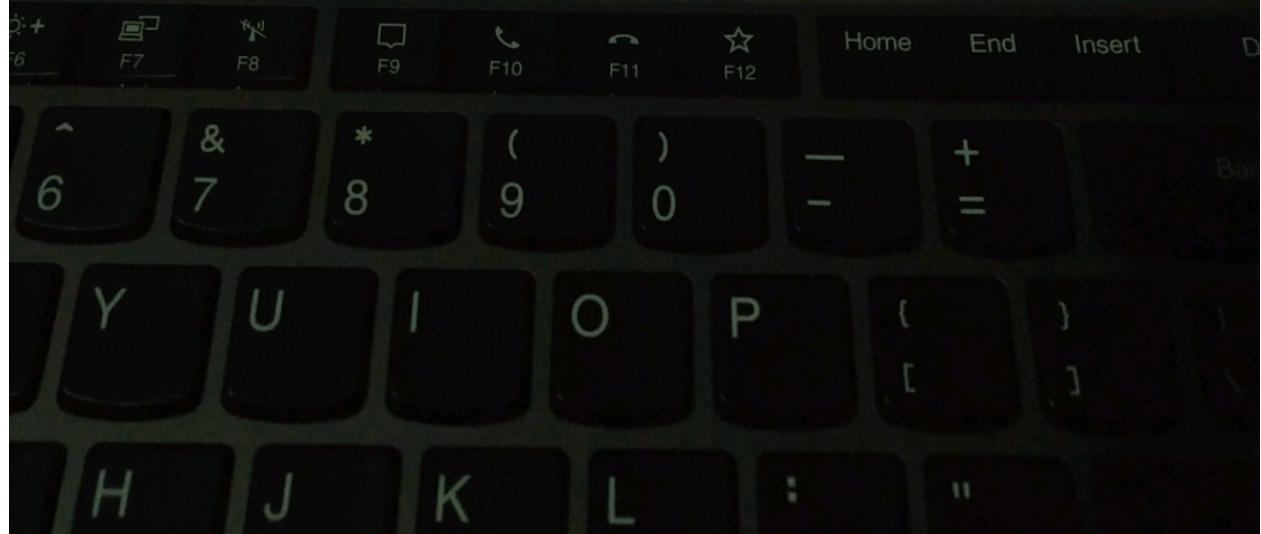
ENG US WiFi 5/2



```
129 }
130 interface CustomerService
131 {
132     void createCustomer(Customer customer);
133     Customer updateCustomer(Customer customer);
134     void deleteCustomer(int id);
135     Customer searchCustomer(int id);
136     Customer[] getCustomers();
137 }
138 class CustomerServiceImpl implements CustomerService
139 {
140     public static Customer customerArray[] = new Customer[5];
141     public static int count = 0;
142     @Override
143     public void createCustomer(Customer customer)
144     {
145         customerArray[count++] = customer;
146     }
147     @Override
148     public Customer updateCustomer(Customer customer) throws CustomerNotFoundException{
149         for(int i=0;i<5;i++)
150         {
151             if(customer.getUserId() == customerArray[i].getUserId())
152                 customerArray[i] = customer;
153             return customerArray[i];
154         }
155     }
156     throw new CustomerNotFoundException();
157 }
158 @Override
```



ENG US 5/2



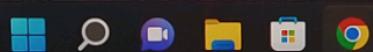
```
146
147     @Override
148     public Customer updateCustomer(Customer customer) throws CustomerNotFoundException{
149         for(int i=0;i<5;i++)
150         {
151             if(customer.getUserId()==customerArray[i].getUserId()){
152                 customerArray[i]=customer;
153                 return customerArray[i];
154             }
155         }
156         throw new CustomerNotFoundException();
157     }
158     @Override
159     public void deleteCustomer(int id) throws CustomerNotFoundException{
160         for(int i=0;i<5;i++)
161         {
162             if(customerArray[i].getUserId()==id)
163             {
164                 customerArray[i]=null;
165             }
166         }
167         throw new CustomerNotFoundException();
168     }
169     @Override
170     public Customer searchCustomer(int id)
171     {
172         for(int i=0;i<5;i++)
173         {
174             if(customerArray[i].getUserId()==id)
175             {
```



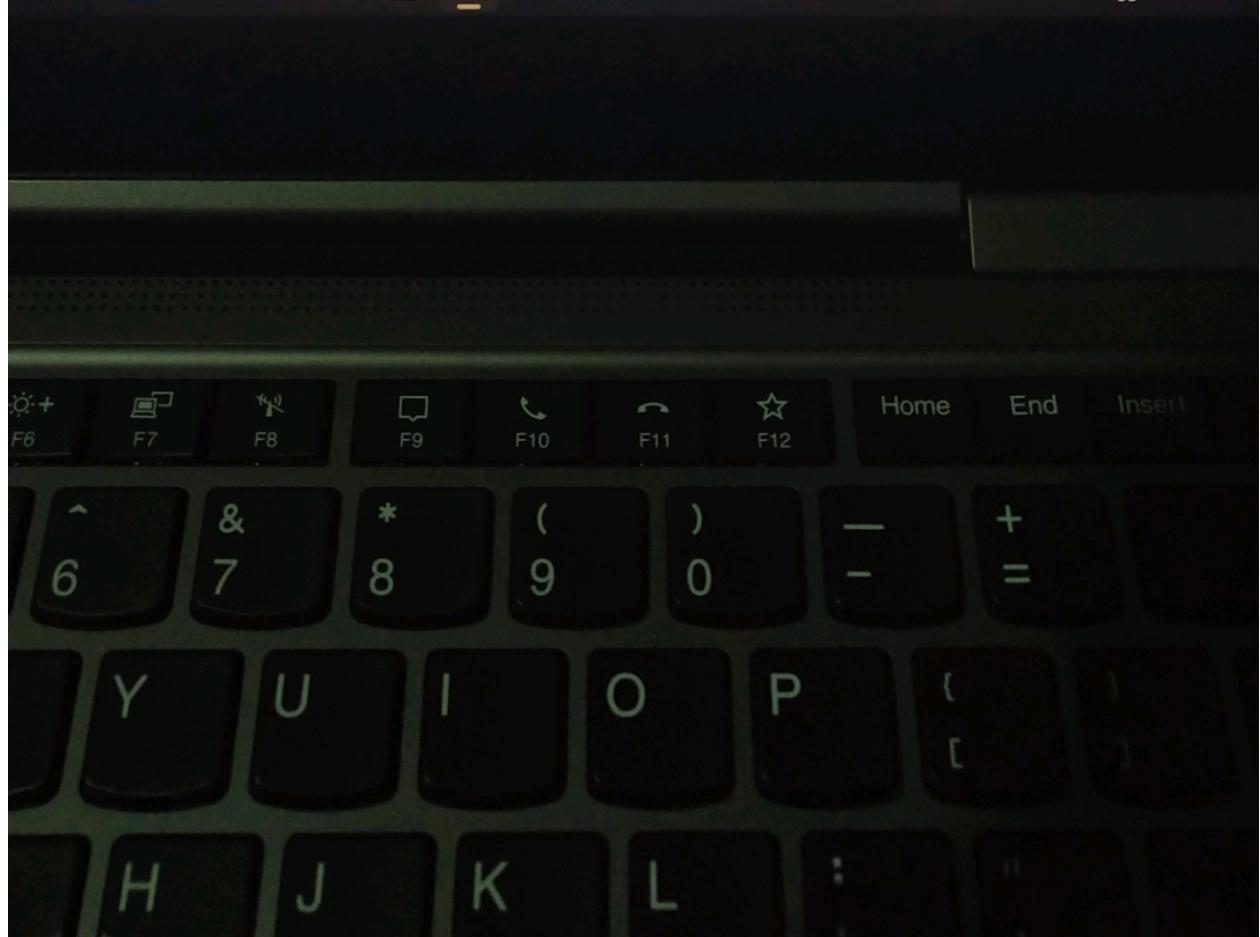
ENG US 9:52 5/28



```
159     public void deleteCustomer(int id) throws CustomerNotFoundException{
160         for(int i=0;i<5;i++)
161     {
162         if(customerArray[i].getUserId()==id)
163     {
164         customerArray[i]=null;
165     }
166     }
167     throw new CustomerNotFoundException();
168 }
169 @Override
170 public Customer searchCustomer(int id)
171 {
172     for(int i=0;i<5;i++)
173     {
174         if(customerArray[i].getUserId()==id)
175     {
176         return customerArray[i];
177     }
178 }
179     throw new CustomerNotFoundException();
180 }
181 @Override
182 public Customer[] getCustomers()
183 {
184     return customerArray;
185 }
186 }
187 class Source{
188     public static void main(String[] args){
```



ENG
US  



```
164             customerArray[i]=null;
165         }
166     }
167     throw new CustomerNotFoundException();
168 }
169 @Override
170 public Customer searchCustomer(int id)
171 {
172     for(int i=0;i<5;i++)
173     {
174         if(customerArray[i].getUserId()==id)
175         {
176             return customerArray[i];
177         }
178     }
179     throw new CustomerNotFoundException();
180 }
181 @Override
182 public Customer[] getCustomers()
183 {
184     return customerArray;
185 }
186 }
187 class Source{
188     public static void main(String[] args){
189
190     }
191 }
```



ENG
US

