Name: Shalin Patel

Id: 38

16242863

Answer 1:

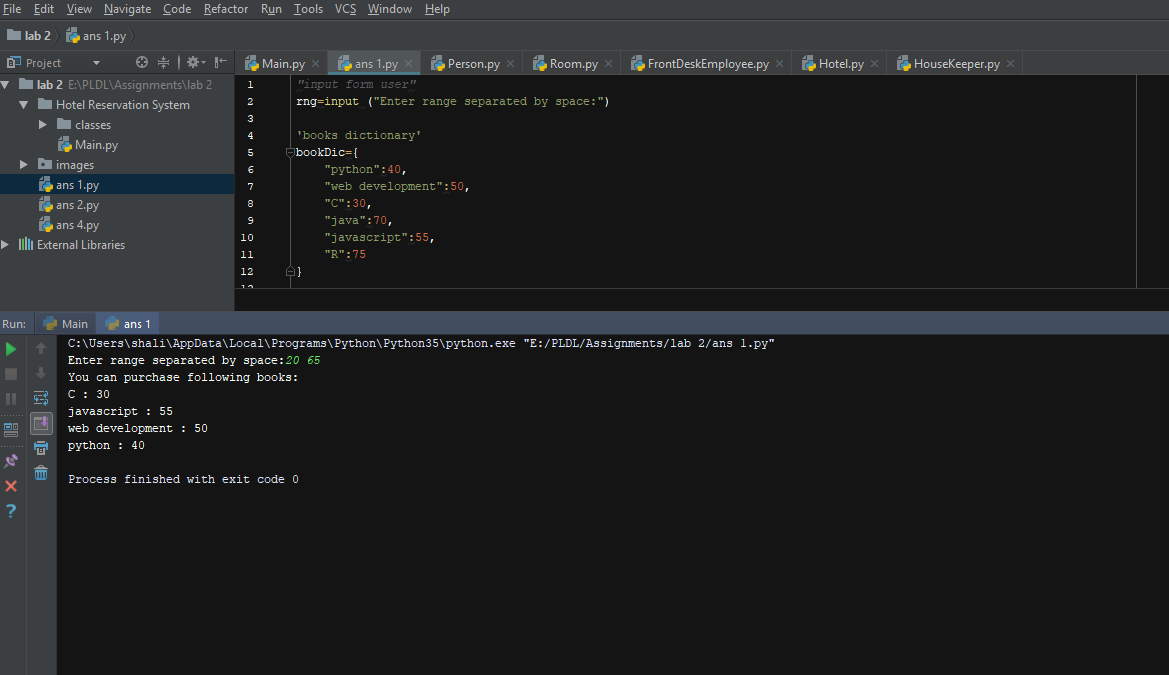
Code

*"input form user"*rng=input ("Enter range separated by space:")  
  
'books dictionary'  
bookDic={  
 "python":40,  
 "web development":50,  
 "C":30,  
 "java":70,  
 "javascript":55,  
 "R":75  
}  
  
'split to get start and end range'  
rng=rng.split(" ")  
  
'get dictionaries in specific range'  
resultDic=dict((k,v) for k, v in bookDic.items() if int(rng[0])<v<int(rng[1]))  
  
'print result'  
print("You can purchase following books:")  
for book in resultDic:  
 print(book,":",resultDic[book])

Generate random integer using numpy

Count most frequent number using bincount

Output



Answer 2:

Code

*"contact list"*Contact\_list=[  
 {"name":"Rashmi","number":"8797989821","email":"rr@gmail.com"},  
 {"name":"Saria","number":"9897989821","email":"ss@gmail.com"},  
 {"name": "shalin", "number": "5214521452", "email": "shalin@gmail.com"},  
 {"name":"patel","number":"9659874632","email":"patel@gmail.com"},  
 {"name":"megha","number":"9853200147","email":"megha@gmail.com"},  
]  
  
prompt="""Choose operation by entering number:  
press a to display contact by name  
press b to display contact by number  
press c to edit contact by name  
press any character to exit   
"""  
  
'function to display contact by name'  
def contactByName():  
 name=input("Enter name to find contact: ")  
 for contact in Contact\_list:  
 if(contact.get('name')==name):  
 print("{name:",contact.get('name'),", number:",contact.get('number'),", email:",contact.get('email'),"}")  
  
'function to display contact by number'  
def contactByNumber():  
 number = input("Enter number to find contact: ")  
 for contact in Contact\_list:  
 if (contact.get('number') == number):  
 print("{name:", contact.get('name'), ", number:", contact.get('number'), ", email:", contact.get('email'),  
 "}")  
  
'function to edit contact by name'  
def editByName():  
 name = input("Enter name to find contact: ")  
 for contact in Contact\_list:  
 if (contact.get('name') == name):  
 number=input("Enter new number: ")  
 contact['number']=number  
 print("Updated contact is {name:", contact.get('name'), ", number:", contact.get('number'), ", email:", contact.get('email'),  
 "}")  
  
"dictionary for operation"  
operationDic={  
 "a":contactByName,  
 "b":contactByNumber,  
 "c":editByName,  
}  
  
  
while(1==1):  
 userInput=input(prompt)  
 if userInput in ["a","b","c"]:  
 operationDic[userInput]()  
 else:  
 break

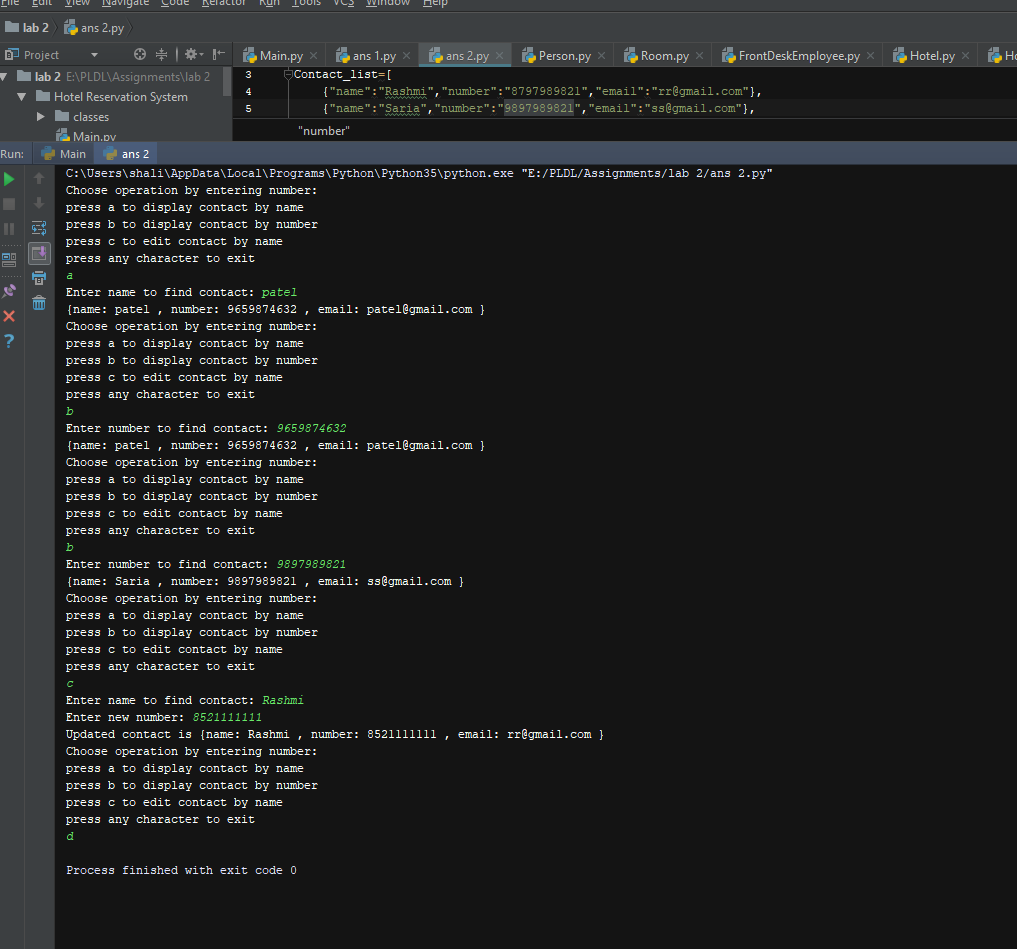
Generated Static list of contact

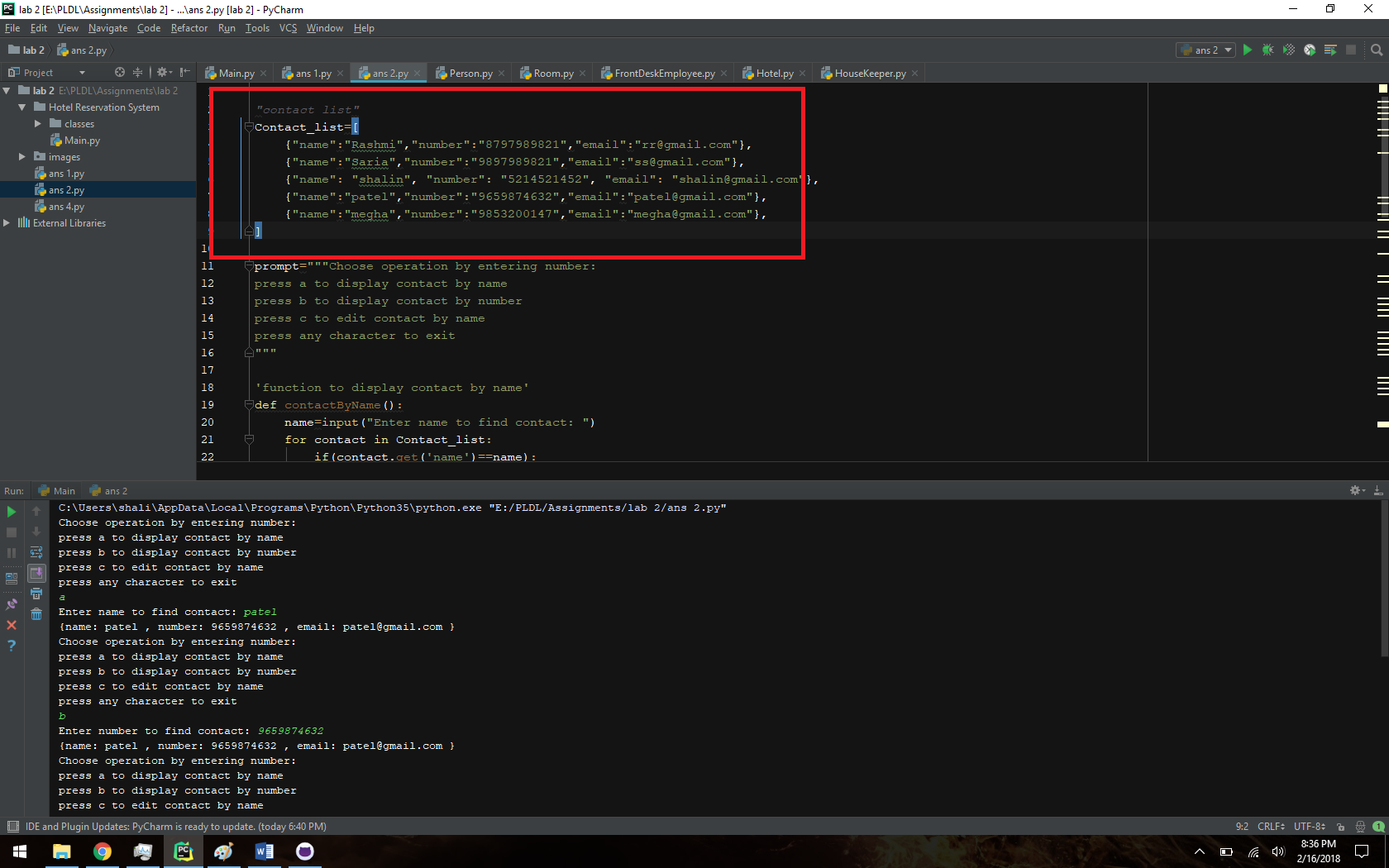
While loop to get option from user

* a to get contact by Name
* b to get contact by Number
* c to get edit by name
* any character to exit

Use of dictionary instead of switch case

Output:





Hotel Reservation system

*#create class Room*class Room(object):  
 def \_\_init\_\_(self,number,roomType,price):  
 self.number=number  
 self.roomType=roomType  
 self.price=price  
  
*#create a class named Hotel*class Hotel(object):  
 *#Static Properties declaration* totalRooms = 76  
 soldRooms = 51  
 availableRooms = 22  
 roomForMaintenance = 3  
 counterBalance=401  
  
 def \_\_init\_\_(self,name,address):  
 self.name=name  
 self.address=address  
 self.houseKeepers={}  
 self.fdEmployee = {}  
 self.rooms=list()  
   
 *#method to register housekeepr* def registerHouseKeep(self,houseKeeper):  
 self.houseKeepers[houseKeeper.name]=houseKeeper  
 print(houseKeeper.name, " is added as a Housekeeper")  
  
 *#method to register FrontDesk Employee* def registerFrontDeskEmployee(self, fdEmployee):  
 self.fdEmployee[fdEmployee.name]=fdEmployee  
 print(fdEmployee.name, " is added as a front desk employee")  
  
 *#method to get Status of motel* def getStatus(self):  
 print("----------------Hotel",self.name ,"Status------------------")  
 print("Total rooms: ",Hotel.totalRooms)  
 print("Sold rooms: ", Hotel.soldRooms)  
 print("Available rooms: ", Hotel.availableRooms)  
 print("Rooms under maintenance: ", Hotel.roomForMaintenance)  
 print("------------------------------------------------------------")  
   
 *#method to register room* def registerRooms(self,room):  
 self.rooms.append(room)  
  
*#create a class named Person*class Person:  
 def \_\_init\_\_(self,name,age,phNo):  
 self.name=name  
 self.age=age  
 self.phNo=phNo  
*#creare class named Customer which inherits Person*class Customer(Person):  
 def \_\_init\_\_(self,name,age,phNo):  
 super().\_\_init\_\_(name,age,phNo)  
  
*#creare class named House Keeper which inherits Person*class HouseKeeper(Person):  
 def \_\_init\_\_(self,name,age,phNo):  
 super().\_\_init\_\_(name,age,phNo)  
 self.roomsAssigned=0  
  
 def assignRooms(self,roomsAssigned):  
 self.roomsAssigned=roomsAssigned  
  
*#creare class Front Desk employee which inherits Person*class FrontDeskEmployee(Person):  
 def \_\_init\_\_(self,name,age,phNo):  
 super().\_\_init\_\_(name, age, phNo)  
 self.selectedRoomNumber=0  
 self.salary=10000  
  
 *# method to open shift* def openShift(self):  
 print("------------------------------------------------------------")  
 print("Shift opened by ",self.name)  
  
 *# method to register Customer* def registerCustomer(self):  
 print("-----------------------Take Customer Details---------------")  
 name=input("Customer Name")  
 age=input("Customer Age")  
 phNo=input("Customer Phone Number")  
 customer=Customer(name,age,phNo)  
 print("Customer ",customer.name ," registered successfully")  
  
 *#* def calculateCounterBalance(self):  
 print("Counter balance is ",Hotel.counterBalance)  
  
 def askForRoom(self,hotel):  
 print("-----------------------Take Room Requirement---------------")  
 roomType=input("What kind of room required? ")  
 price= int(input("Price? "))  
 rooms=list()  
   
 *#get rooms that matches requirement* for room in hotel.rooms:  
 if room.price==price or room.roomType==roomType :  
 rooms.append(room)  
 print("Available rooms are")  
 for room in rooms:  
 print("Room Number: ",room.number," Room type: ",room.roomType," Price: ",room.price)  
  
 def bookRoom(self):  
 selectedRoomNumber=int(input("Which room you want to book? "))  
   
 *#book room by selected customer* for room in hotel.rooms:  
 if room.number == selectedRoomNumber:  
 print("Congrats!!!!")  
 print("You booked Room Number ",room.number," Room Type ",room.roomType," Price ",room.price)  
 Hotel.availableRooms= Hotel.soldRooms- 1  
 Hotel.soldRooms= Hotel.soldRooms+1  
 Hotel.counterBalance = Hotel.counterBalance+room.price  
 break  
  
 def closeShift(self):  
 print("Shift Closed")  
  
*#Hotel Registartion and Setup*hotel=Hotel("Quality Inn","5100 Cherry St,kcmo, 64109")  
hotel.getStatus()  
  
room1=Room(101,"Double bed",70)  
hotel.registerRooms(room1)  
room2=Room(201,"Queen bed",50)  
hotel.registerRooms(room2)  
room3=Room(301,"King bed",60)  
hotel.registerRooms(room3)  
room4=Room(102,"King bed",60)  
hotel.registerRooms(room4)  
room5=Room(103,"Double bed",70)  
hotel.registerRooms(room5)  
room6=Room(202,"Queen bed",50)  
hotel.registerRooms(room6)  
  
houseKeep1=HouseKeeper("Bharat","21","8512541236")  
houseKeep1.roomsAssigned=15  
  
houseKeep2=HouseKeeper("Raju","35","9856325412")  
houseKeep2.roomsAssigned=32  
  
hotel.registerHouseKeep(houseKeep1)  
hotel.registerHouseKeep(houseKeep2)  
  
  
fdEmplyee1=FrontDeskEmployee("Vinita","30","8163288464")  
fdEmplyee2=FrontDeskEmployee("Jinal","26","8511318649")  
  
hotel.registerFrontDeskEmployee(fdEmplyee1)  
hotel.registerFrontDeskEmployee(fdEmplyee2)  
  
*#Shift oper*fdEmplyee1.openShift()  
*#callculate Balance*fdEmplyee1.calculateCounterBalance()  
  
*# get customer info and register customer*fdEmplyee1.registerCustomer()  
  
*#Adsk for room and book room*fdEmplyee1.askForRoom(hotel)  
fdEmplyee1.bookRoom()  
fdEmplyee1.calculateCounterBalance()  
  
*#show hotel status*hotel.getStatus()

Setup Hotel by adding its name, address.

Register rooms, House keeper, Front desk employee

FrontdeskEmplooye register customer

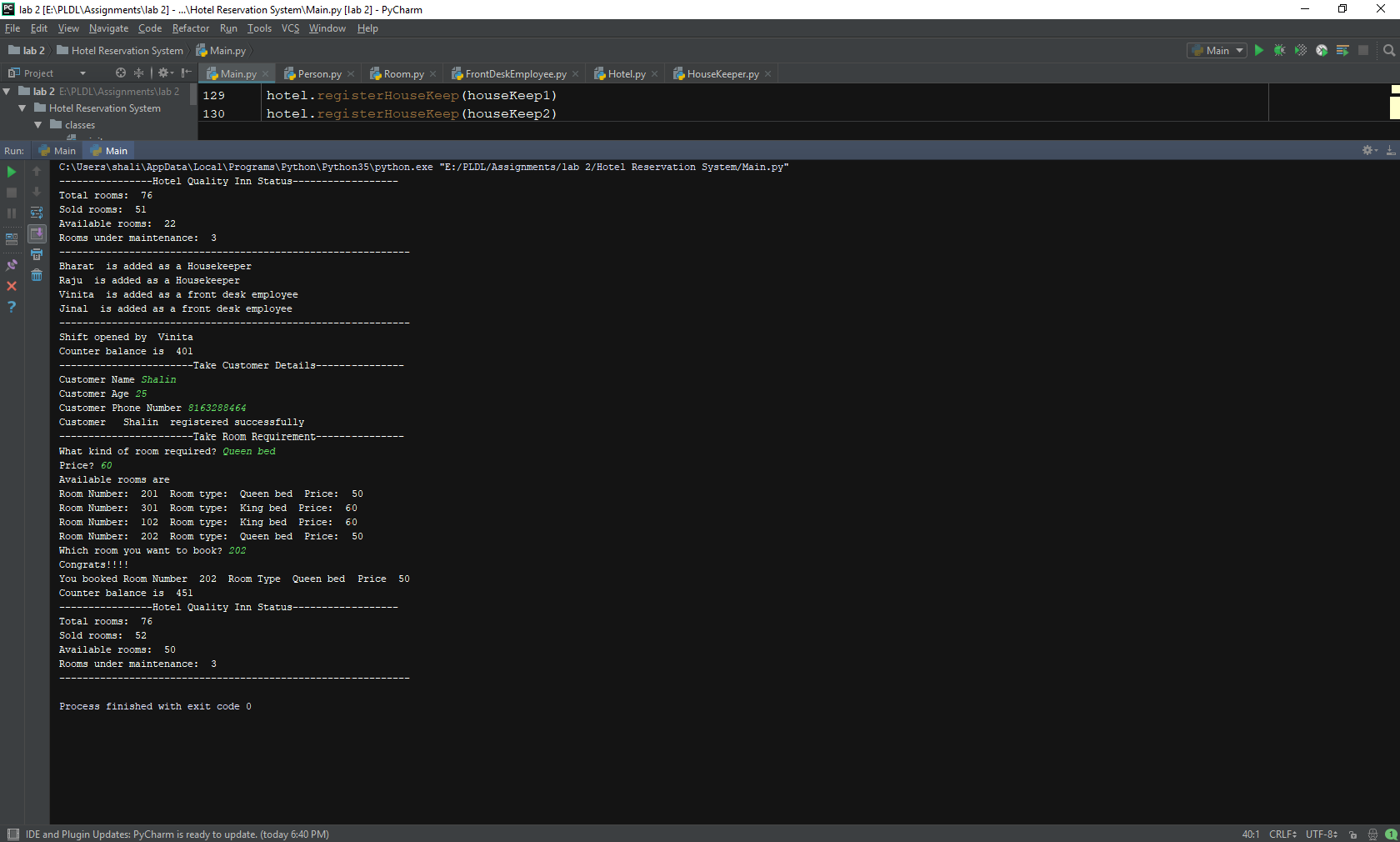
Ask for room

Customer selected room

Book Room

Update Status in Hotel

Output:



Answer 4:

import numpy as np  
  
'generate an array of int in range 1 to 20'  
arr=np.random.randint(1,20,15,int)  
  
'print array'  
print("Generated Array:")  
print(arr)  
print("Most frequent number is")  
  
'count most frequent number'  
counts = np.bincount(arr)  
print(np.argmax(counts))

Output:

