Sravani Konujula

Class ID-23

16230172

Lab Assignment-2

**Objective:**

The purpose of this assignment is

* To create UMKC books list and print books between price range
* To display a contact list with edit and display options
* To use classes in library management system
* To display a highest frequency number from random numbers.

**Features:**

* In bookshop with dictionary order it takes the minimum and maximum prices range and display in between ranges.
* Initially takes the contact details in a list and uses the given options to display and edit contact and numbers.
* Using numpy display a highest frequency number from random numbers.

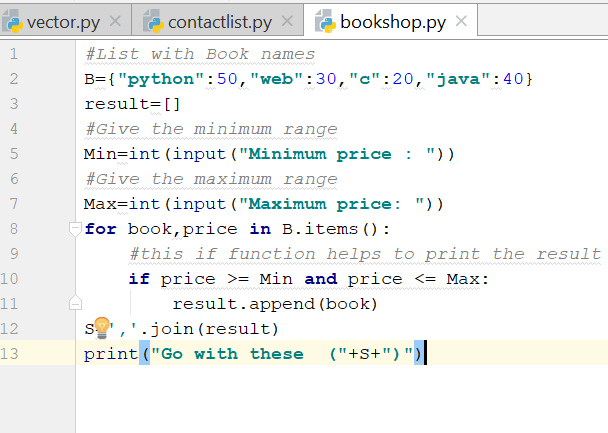
**Configuration:**

Pycharm

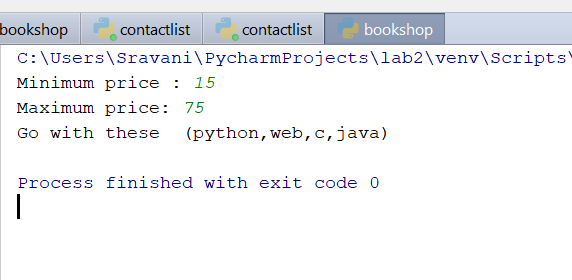
Python: 2.7.13

**Output Screens**

**Question1:**

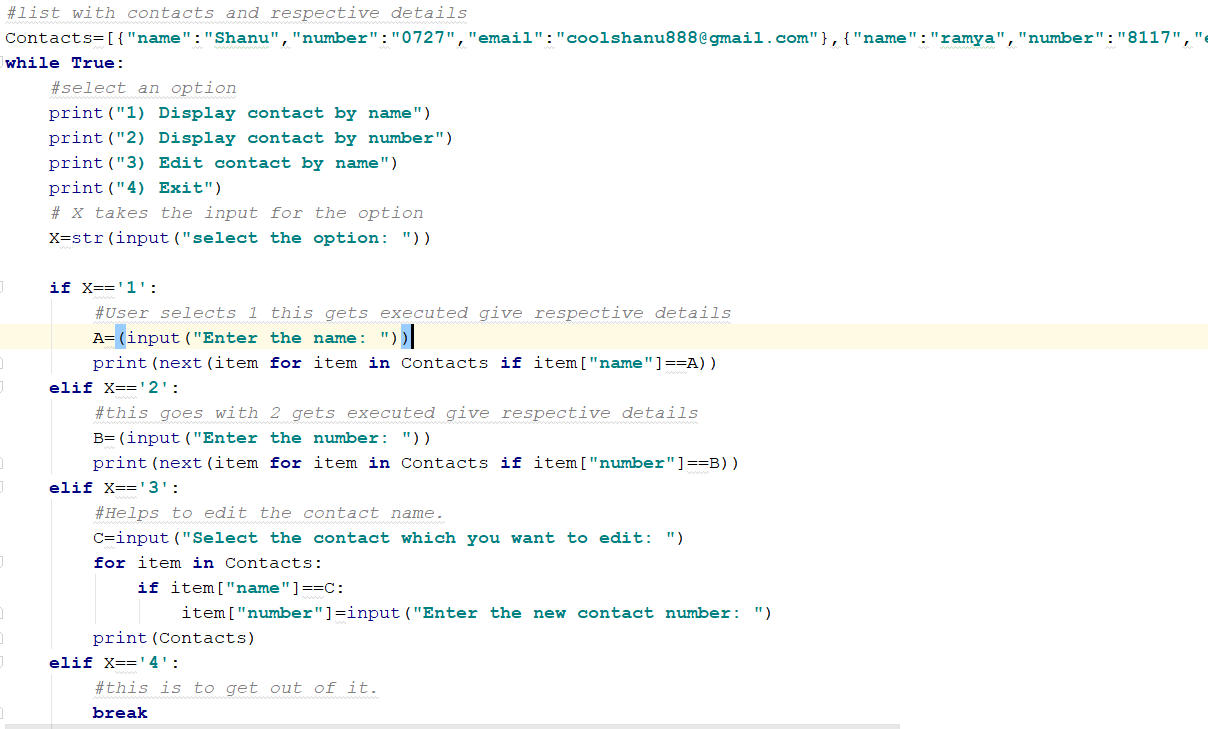


**Output:**



**Question2:**

**Source Code:**

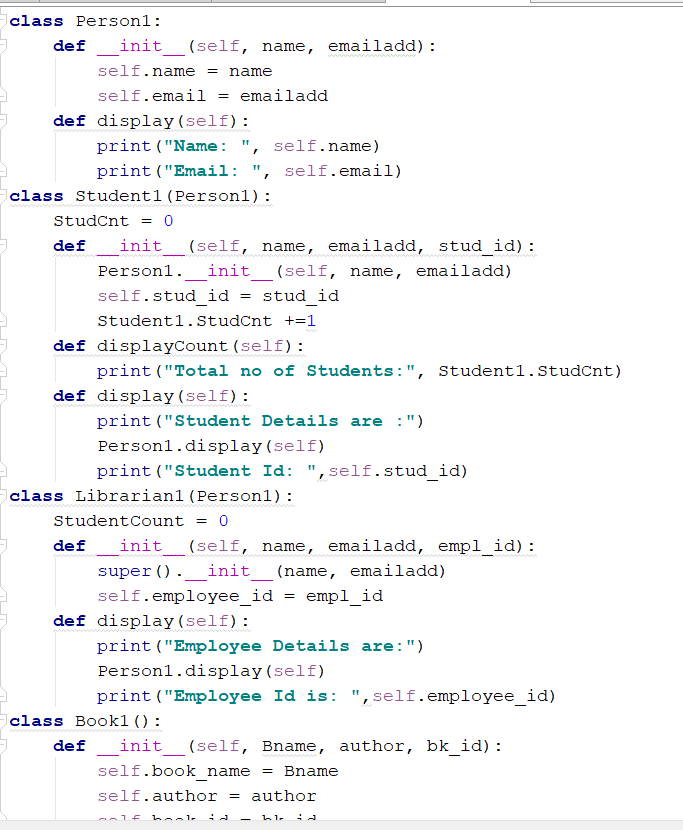


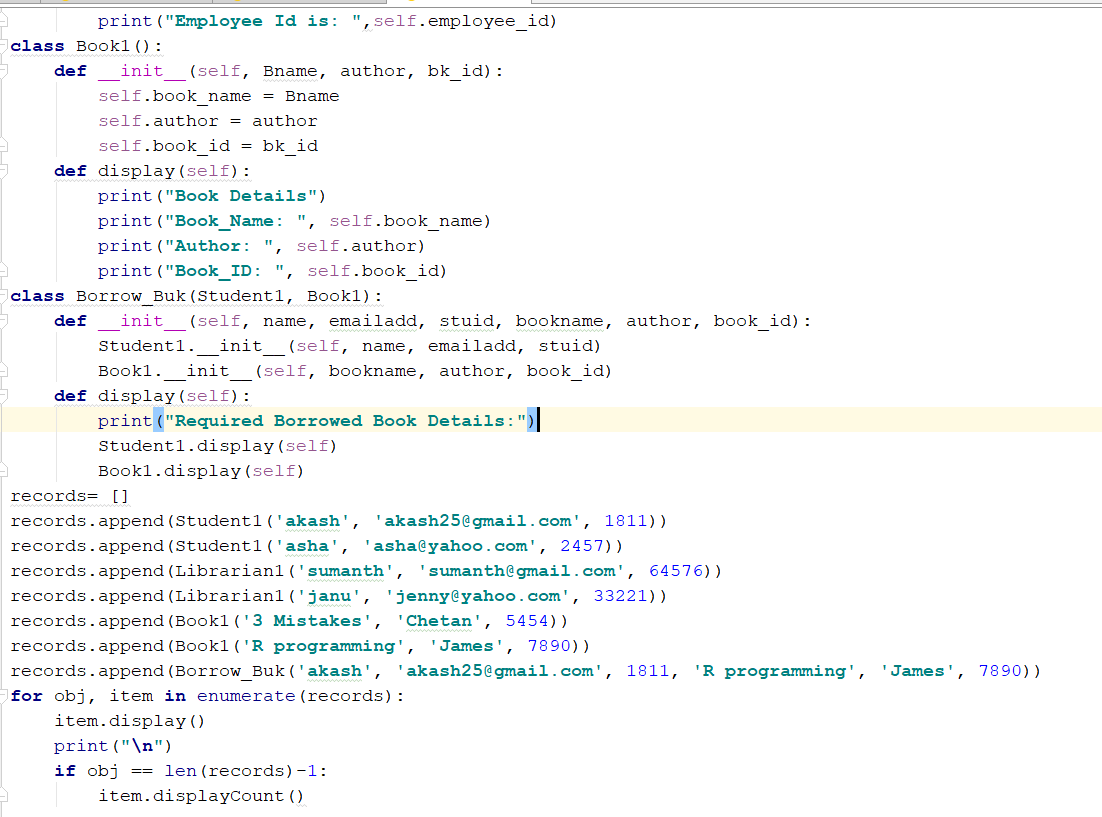
**Output Screen:**



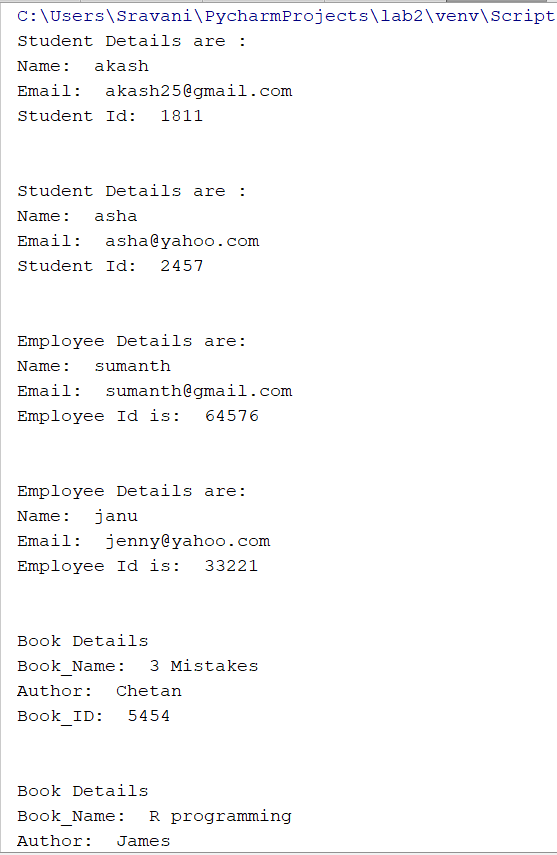
**Question3:**

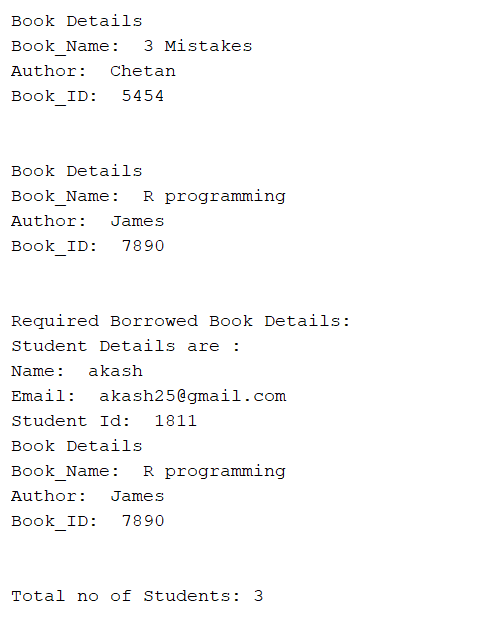
**Source Code:**





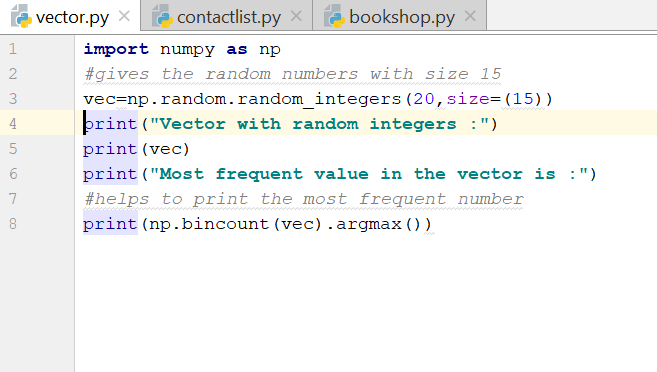
**Output Screen:**



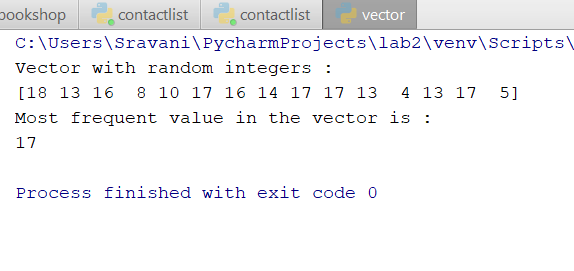


**Question4:**

**Source Code:**



**Output Screen:**



**Code Snippet1**

*#List with Book names*B={**"python"**:50,**"web"**:30,**"c"**:20,**"java"**:40}  
result=[]  
*#Give the minimum range*Min=int(input(**"Minimum price : "**))  
*#Give the maximum range*Max=int(input(**"Maximum price: "**))  
**for** book,price **in** B.items():  
 *#this if function helps to print the result* **if** price >= Min **and** price <= Max:  
 result.append(book)  
S=**','**.join(result)  
print(**"Go with these ("**+S+**")"**)

**Code Snippet2**

*#list with contacts and respective details*Contacts=[{**"name"**:**"Shanu"**,**"number"**:**"0727"**,**"email"**:**"coolshanu888@gmail.com"**},{**"name"**:**"ramya"**,**"number"**:**"8117"**,**"email"**:**"ramyakonujula@gmail.com"**},{**"name"**:**"Sravani"**,**"number"**:**"8111"**,**"email"**:**"sravanikonujula@gmail.com"**}]  
**while True**:  
 *#select an option* print(**"1) Display contact by name"**)  
 print(**"2) Display contact by number"**)  
 print(**"3) Edit contact by name"**)  
 print(**"4) Exit"**)  
 *# X takes the input for the option* X=str(input(**"select the option: "**))  
  
 **if** X==**'1'**:  
 *#User selects 1 this gets executed give respective details* A=(input(**"Enter the name: "**))  
 print(next(item **for** item **in** Contacts **if** item[**"name"**]==A))  
 **elif** X==**'2'**:  
 *#this goes with 2 gets executed give respective details* B=(input(**"Enter the number: "**))  
 print(next(item **for** item **in** Contacts **if** item[**"number"**]==B))  
 **elif** X==**'3'**:  
 *#Helps to edit the contact name.* C=input(**"Select the contact which you want to edit: "**)  
 **for** item **in** Contacts:  
 **if** item[**"name"**]==C:  
 item[**"number"**]=input(**"Enter the new contact number: "**)  
 print(Contacts)  
 **elif** X==**'4'**:  
 *#this is to get out of it.* **break**

**Code Snippet3:**

**class** Person1:  
 **def** \_\_init\_\_(self, name, emailadd):  
 self.name = name  
 self.email = emailadd  
 **def** display(self):  
 print(**"Name: "**, self.name)  
 print(**"Email: "**, self.email)  
**class** Student1(Person1):  
 StudCnt = 0  
 **def** \_\_init\_\_(self, name, emailadd, stud\_id):  
 Person1.\_\_init\_\_(self, name, emailadd)  
 self.stud\_id = stud\_id  
 Student1.StudCnt +=1  
 **def** displayCount(self):  
 print(**"Total no of Students:"**, Student1.StudCnt)  
 **def** display(self):  
 print(**"Student Details are :"**)  
 Person1.display(self)  
 print(**"Student Id: "**,self.stud\_id)  
**class** Librarian1(Person1):  
 StudentCount = 0  
 **def** \_\_init\_\_(self, name, emailadd, empl\_id):  
 super().\_\_init\_\_(name, emailadd)  
 self.employee\_id = empl\_id  
 **def** display(self):  
 print(**"Employee Details are:"**)  
 Person1.display(self)  
 print(**"Employee Id is: "**,self.employee\_id)  
**class** Book1():  
 **def** \_\_init\_\_(self, Bname, author, bk\_id):  
 self.book\_name = Bname  
 self.author = author  
 self.book\_id = bk\_id  
 **def** display(self):  
 print(**"Book Details"**)  
 print(**"Book\_Name: "**, self.book\_name)  
 print(**"Author: "**, self.author)  
 print(**"Book\_ID: "**, self.book\_id)  
**class** Borrow\_Buk(Student1, Book1):  
 **def** \_\_init\_\_(self, name, emailadd, stuid, bookname, author, book\_id):  
 Student1.\_\_init\_\_(self, name, emailadd, stuid)  
 Book1.\_\_init\_\_(self, bookname, author, book\_id)  
 **def** display(self):  
 print(**"Required Borrowed Book Details:"**)  
 Student1.display(self)  
 Book1.display(self)  
records= []  
records.append(Student1(**'akash'**, **'akash25@gmail.com'**, 1811))  
records.append(Student1(**'asha'**, **'asha@yahoo.com'**, 2457))  
records.append(Librarian1(**'sumanth'**, **'sumanth@gmail.com'**, 64576))  
records.append(Librarian1(**'janu'**, **'jenny@yahoo.com'**, 33221))  
records.append(Book1(**'3 Mistakes'**, **'Chetan'**, 5454))  
records.append(Book1(**'R programming'**, **'James'**, 7890))  
records.append(Borrow\_Buk(**'akash'**, **'akash25@gmail.com'**, 1811, **'R programming'**, **'James'**, 7890))  
**for** obj, item **in** enumerate(records):  
 item.display()  
 print(**"\n"**)  
 **if** obj == len(records)-1:  
 item.displayCount()

**Code Snippet4:**

**import** numpy **as** np  
*#gives the random numbers with size 15*vec=np.random.random\_integers(20,size=(15))  
print(**"Vector with random integers :"**)  
print(vec)  
print(**"Most frequent value in the vector is :"**)  
*#helps to print the most frequent number*print(np.bincount(vec).argmax())

**Deployment:**

Code is written in python and we used pycharm to run this and printed result in the python console.

**Limitations:**

There are no limitations for the executed code snippets.