Vania Todorova

CS 490

Lab2

https://github.com/vanetoj/Lab2-CS5590

Ex 1

"""the customer enters "from- to" range of price and then the for loop iterates trough dict that has for keys the names of the books and the prices for the values. if it falls into the range given by the customer the book gets appended to a list called canbuylist"""

```
price_from = int(input("Enter price from: "))
price_to = int(input("Enter price to: "))
```

```
canbuylist = []

for k, v in books.items():
    if (v >= price_from and v <= price_to):
        canbuylist.append(k)

print("You can purchase: ", canbuylist)
```

Ex 2

"""we have all the contacts in a list of dictionaries where each dictionary stores the info of an individual person- name, number , email. The user is asked for choices to enter a name or a number or to edit . If loops give choices to display the contact or to edit the number through an iteration of the dictionaries in the list"""

```
contact_list = [ { "name" : "Rashmi" ,
                   "number" : "8797989821" ,
                   "email" : "rr@gmail.com" },
                 { "name" : "Saria" ,
                   "number" : "9897988245" ,
                   "email" : "ss@gmail.com" } ]


enter_a_choice=input("please enter 1 to find by name or 2 to find by number and 3 if you
want to edit any of the numbers ")
if enter_a_choice=='1':
    get_name=input(" Enter a name ")
if get_name=="Rashmi":
    print(contact_list[0])
if get_name=="saria":
    print(contact_list[1])
```

```
if enter_a_choice=="2":
    get_number=input("enter a number")
if get_number=='8797989821':
    print(contact_list[0])
elif get_number==9897988245:
    print(contact_list[1])

if enter_a_choice=='3':
    name_to_edit=input("which number would you like to edit? ")
if name_to_edit=="rashmi":
    get_new_number=int(input("Enter the new number "))
    for d in contact_list:
        d.update((k, get_new_number) for k, v in d.items() if v == "8797989821")
elif name_to_edit=="saria":
    get_new_number=int(input("enter the new number "))
    for d in contact_list:
        d.update((k, get_new_number) for k, v in d.items() if v == "9897988245")
print(contact_list)
```

Ex 3

"""The super class is Book that has the 2 methods: init with arguments book name, isbn and author and a method to display the book with the arguments. Then we have class Library that has the list of the books in the library. The library class also has checkin and check out methods to keep track which book is in or out of the library.It removes the ones that are checked out to keep the available list current.We also have class person with 2 subclasses - student and librarian. Sub class assistant librarian class is an example of multiple inheritence as well as it has a super call in it."""

```
class Book:
    def __init__(self, book_name, isbn, author):
        self.name = book_name
        self.isbn = isbn
        self.author = author

    def display(self):
        print("  Book: ", self.name)
        print("  ISBN: ", self.isbn)
        print("  Author", self.author)


class Library:
    booklist = [
        Book("The Hobbit", "10", "J.R.R.R.R. Tolkien"),
        Book("Learn Python", "11", "Albert Snek"),
        Book("Learn Java", "12", "Bill Gates")]

    def __init__(self):
        self.__books = Library.booklist

    def checkIn(self, student, book):
```

```
        self.__books.append(book)
        student.hecked_out_books.remove(book)
        print(student.name, " checked in ", book.name)

    def checkOut(self, student, isbn):
        for book in self.__books:
            if book.isbn == isbn:
                print(student.name, " checked out ", book.name)
                self.__books.remove(book)
                student.hecked_out_books.append(book)

    def display(self):
        print("Books in library: ")
        for book in self.__books:
            book.display()
            print()


class Person:
    def __init__(self, person_name):
        self.name = person_name

    def display(self):
        print("Name: ", self.name)


class Student(Person):
    def __init__(self, name, student_id):
        Person.__init__(self, name)
        self.student_id = student_id
        self.checked_out_books = []

    def display(self):
        Person.display(self)
        print("Student ID: ", self.student_id)
        print("Has Books: ")
        for book in self.checked_out_books:
            book.display()
            print()


class Librarian(Person):
    def __init__(self, name, employee_id):
        Person.__init__(self, name)
        self.employee_id = employee_id

    def display(self):
        Person.display(self)
        print("Employee ID: ", self.employee_id)


class AssistantLibrarian(Student, Librarian):
    def __init__(self, name, student_id, employee_id):
        Student.__init__(self, name, student_id)
        Librarian.__init__(self, name, employee_id)
```

```
    def display(self):
        Student.display(self)
        Librarian.display(self)


person = Person("Just Someone")
person.display()
print()

student = Student("Vania", "S-14524")
student.display()
print()

librarian = Librarian("Mildred", "E-12342")
librarian.display()
print()

assistant = AssistantLibrarian("Jack", "S-111", "E-5123")
assistant.display()
print()

library = Library()
library.display()
print()

library.checkOut(student, "11")
library.checkOut(student, "12")
library.display()
student.display()

library.checkIn(student, student.hecked_out_books[0])
library.display()
student.display()
```

Ex 4

""" First you install the numpy package . then you set the random function to create a vector. The first 2 arguments indicate the range and the third the size of the vector. The I used the built in numpy function bincount to find the most frquent value ."""

```
import numpy as np
x=np.random.randint(0,20,15)
print("The original array is:",x)
print ("The most frequent value is : ")
print(np.bincount(x).argmax())
```