Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
BCA Semester – V
CA329 Introduction to Multi Paradigm Programming Language
Sessional Practical Examination

Date:11/10/2024

Q1. Write a Python to perform the below operation to Library Management with given data.

Library= {‘BK001’: {‘BookDetails’: [‘python’,’Nov-20’,790,’Basic and advance python’], ’AuthorDetails’: [‘guido Van’] },

‘BK102’: {‘BookDetails’: [‘Java’,’Jun -18’,790,’servlet in java’], ’AuthorDetails’: [‘Ivan Bayross’,’cynthia Bayross’] },

}

Add 3 more random book records statically in the dictionary.

Now Perform Below Operations with user choice in while loop

- **Add new book record with user input ->** ask user to enter book details and author details and Add it in the dictionary in same dictionary structure. [5]
- **Modify the book price based on the book id ->** Ask user to enter book id to change their price. [3]
- **Remove the Author Name ->** Ask user to enter book id and author name to delete if exist. Otherwise show message that author is not matching to book. [3]
- **Search Option->** user can search the book with help of book description with any key and list out the available books. [5]
- **Print the All book details with author details with good format.** [4]

Q2. Write a Python to perform the below operation to Library Management with given data.

```
Library = {  'BK001':
  { 'BookDetails': { 'Title': 'Python', 'Published': 'Nov-20', 'Price': 790, 'Description': 'Basic and
advanced Python' },
'AuthorDetails': { 'PrimaryAuthor': 'Guido Van Rossum', 'CoAuthors': [] },
'StockDetails': { 'AvailableCopies': 10, 'TotalCopies': 50, 'IsInStock': True }
},
'BK102': { 'BookDetails': { 'Title': 'Java', 'Published': 'Jun-18', 'Price': 820, 'Description': 'Servlet in
Java' },
'AuthorDetails': { 'PrimaryAuthor': 'Ivan Bayross', 'CoAuthors': ['Cynthia Bayross'] },
'StockDetails': { 'AvailableCopies': 0, 'TotalCopies': 25, 'IsInStock': False }
 },

 }
```

Now Perform Below Operations with user choice in while loop

- **Selling the book ->** ask user to enter id to sell it if available copies is not 0 then it will reduce the available copies by 1. If the available copies become 0, the book will be marked as out of stock (IsInStock = False) [5]
- **Purchase a New Book ->** Add New purchased book into the dictionary structure with Auto generated book id and ask other required details from the user. [5]
- **Restock a existing Book->** Ask user to enter book id to restock a book, which will increase the available copies and mark the book as in stock (IsInStock = True) [3]
- **Check Stock Status->** The user can check whether a book is in stock or out of stock by each book. [4]
- **Display Books All details ->** all the book and author details will be displayed. [3]

Q3. Write a Python to perform the below operation to online Shopping with given data.

Products = { 'P001': { 'Name': 'Smartphone', 'Category': 'Electronics', 'Price': 500, 'Description': 'Latest 5G smartphone with 128GB storage', 'Stock': 30 },

'P002': { 'Name': 'Microwave', 'Category': 'Home Appliances', 'Price': 150, 'Description': '800W Microwave with touch control', 'Stock': 20 },

}
Cart={}

Add 3 more random product records statically in the products dictionary.

View All the Products All Details with availability of product. [4]

Now Perform Below Operations with user choice in while loop

- **Add to Cart**: Prompt the user to enter one or more product IDs to add to the shopping cart. The user can input multiple product IDs at once and **store into cart dictionary with quantity** [4]
- **View Cart ->** User will view cart and Total of all items and Total Bill. [4]
  e.g. product Name, Price, Quantity, Total Amount

  ```
  Smartphone   500   2          1000
  Microwave    150   3           450
                  Total Bill     1450
  ```
- **Remove Product from cart ->** User can remove product item from cart by entering the product id. [4]
- **Buy from the cart->** Total bill amount will be generated, Cart will be empty now, quantity will be reduced from product dictionary and all the products will be displayed again with availability. [4]

Q4. Write a Python to perform the below operation to online Shopping with given data for products and design own data structure for the review and rating of the products given by the multiple user.

Products = { 'P001': { 'Name': 'Smartphone', 'Category': 'Electronics', 'Price': 500, 'Description': 'Latest 5G smartphone with 128GB storage', 'Stock': 30 },

'P002': { 'Name': 'Microwave', 'Category': 'Home Appliances', 'Price': 150, 'Description': '800W Microwave with touch control', 'Stock': 20 },

}

Review_rating={} [4]

Add 3 random review and rating for the above products statically in the dictionary.

Now Perform Below Operations with user choice in while loop

- **Add New Product ->** Ask user to enter all required product details to make same data structure as above to add it as new element into the products dictionary. [4]
- **Add Review and Rating->** Ask users to give review and rating based on the products id and store it on the review_rating dictionary. [4]
- **Display All the products details with Review and rating of all user**. [4]
- **Generate reports for the bad reviewed and rated products details.** [4]

Q5. Write a Python to perform the below operation to manage weekly expense with given data.

```
expenses = {
   'Monday': [
      {'Amount': 50, 'Category': 'Groceries', 'Description': 'Bought vegetables'},
      {'Amount': 20, 'Category': 'Transport', 'Description': 'Bus fare'},
      {'Amount': 200, 'Category': 'Entertainment', 'Description': ‘Netflix Subscription'}
   ],
   'Wednesday': [
      {'Amount': 30, 'Category': 'Transport', 'Description': 'Bus fare'}
       {'Amount': 30, 'Category': 'Entertainment', 'Description': 'Movie ticket'},

   ]
}
```

Now Perform Below Operations with user choice in while loop

- **Add Expense on Current Day by user input ->** As of current data structure current day is Wednesday [4]
- **Start adding expenses for new day(Thursday) with one single expense.** [4]
- **Generate reports of expenses categories wise.**
  ```
  e.g   Entertainment -> 200+30 -> 230
          Transport        -> 20+30 -> 50
           Groceries       ->  50       ->
  ```
  [5]
- **Search Expense based on the expense description->** User will enter the search key and based on the it shows day and amount [5]
- **Edit any expense description of current day** [2]

Q6. Write a Python to perform the below operation to movie ticket management.

```
pvr_movies = {

'Inception': { 'Showtimes': ['10:00 AM', '2:00 PM', '6:00 PM'],
                 'Price': 10, ‘Categoty’:’action’,’Capacity’:150, ‘status’:’on’ },
'Interstellar': { 'Showtimes': ['11:00 AM', '3:00 PM', '7:00 PM'],
                 'Price': 12, ‘Categoty’:’ action’,’Capacity’:100, ‘status’:’on’ },
'The Dark Knight': { 'Showtimes': ['12:00 PM', '4:00 PM', '8:00 PM'],
                       'Price': 15, ‘Categoty’:’crime’,’Capacity’:200, ‘status’:’off’
 }

}
```

Add 3 random statically movies records in the dictionary.

Now Perform Below Operations with user choice in while loop

- **Add New Movie ->** Ask user to enter Movie name, showtime slots, price, movie category and capacity of screen, make same data structure as above and add into the dictionary. [4]
- **View Available Movies**: Show the list of available movies based on the status and their showtimes. [4]
- **Update movie status:** Ask user if they want to change movie status on to off. [4]
- **Remove the showtime slot:** Ask user to which showtime slot they don’t want to run now and delete it from the dictionary. [4]
- **Display Movies name based on the category.**
  e.g action -> Inception
  -> Interstellar
  Crime -> The Dark Knight [4]