Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
Institute of Computer Technology

Name: Tushar Panchal

En.No: 21162101014

Sub: FP(Functional Programming)

Branch: CBA

Batch:41

----------PRACTICAL 9----------

- **Question-1 :**

Design a library application based on the following requirements:

- Each book in the library has related title and author.
- All the books in library are soft-books which imply that book may exist as either an EBook or Audio book.
- Every EBook has: Format: PDF, EPUB, MOBI, AZW. Anything else supplied as format should give compilation error.
- Every EBook has number of pages.
- Audiobook has: Track length measured in minutes. Format: MP3, WMA, WAV. Anything else supplied as format should give compilation error.
- Application should be able to display books' name, author, format and description if required.

✓ **Source Code :**

```
class Book:
    def __init__(self, book, author):
        self.title = book.split('.')[0]
        self.b_format = book.split('.')[1]
        self.author = author
        self.type = ""


class EBook(Book):
    def __init__(self, book, author, pages):
        super().__init__(book, author)
        self.pages = pages
        self.type = "EBook"
```

```
        if self.b_format not in ['pdf', 'epub', 'mobi', 'azw']:
            raise Exception("Invalid Format!")

    def show_details(self):
        print("\n----------------------------------")
        print(f"\nBook Name : {self.title}")
        print(f"\nBook Author : {self.author}")
        print(f"\nBook Type : {self.type}")
        print(f"\nBook Format : {self.b_format}")
        print(f"\nBook Length : {self.pages}pgs")
        print("\n----------------------------------")


class AudioBook(Book):
    def __init__(self, book, author, length):
        super().__init__(book, author)
        self.length = length
        self.type = "AudioBook"

        if self.b_format not in ['mp3', 'wma', 'wav']:
            raise Exception("Invalid Format!")

    def show_details(self):
        print("\n**********************************")
        print(f"\n  Book Name : {self.title}       ")
        print(f"\n  Book Author : {self.author}    ")
        print(f"\n  Book Type : {self.type}        ")
        print(f"\n  Book Format : {self.b_format}  ")
        print(f"\n  Book Length : {self.length}    ")
        print("\n**********************************")


book1 = EBook('BOOK1.pdf', 'author1', '250')
book2 = AudioBook('BOOK2.mp3', 'Author2', '7:10:25')
# book3 = EBook('BOOK3.mp4', 'Author3', '1:00:00')  # ERROR

book1.show_details()
book2.show_details()
# book3.show_details()
```

✓ **Output :**

```
┌─tushar@tushar in ~/Documents/FP/9 via  v3.10.10
└─λ python 1.py

---------------------------------
Book Name : BOOK1
Book Author : author1
Book Type : EBook
Book Format : pdf
Book Length : 250pgs
---------------------------------
*********************************

 Book Name : BOOK2
 Book Author : Author2
 Book Type : AudioBook
 Book Format : mp3
 Book Length : 7:10:25
*********************************
```

✓ **Question-2 :**

Implement Stack using concept of Object oriented programming.

At minimum, any stack should be able to perform the following three operations:

- Push: Add an object passed as an argument to the top of the stack.
- Pop: Remove the object at the top of the stack and return it.
- Peek (or peep): Return the object at the top of the stack (without removing it).
- Display: Print the current status of stack.

✓ **Source Code :**

```python
class Stack:
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        if not self.is_empty():
            return self.items.pop()
```

```
    def peek(self):
        if not self.is_empty():
            return self.items[-1]

    def is_empty(self):
        return len(self.items) == 0

    def display(self):
        print("Stack: ", self.items)

s = Stack()

while True:
    print("****************")
    print("|  1. Push     |")
    print("|  2. Pop      |")
    print("|  3. Peek     |")
    print("|  4. Display  |")
    print("|  5. Quit     |")
    print("****************")

    choice = int(input("Enter your choice: "))

    if choice == 1:
        item = input("Enter the item to be pushed: ")
        s.push(item)
    elif choice == 2:
        if s.is_empty():
            print("Stack is empty!")
        else:
            print("Popped item: ", s.pop())
    elif choice == 3:
        if s.is_empty():
            print("Stack is empty!")
        else:
            print("Top item: ", s.peek())
    elif choice == 4:
        s.display()
    elif choice == 5:
        break
    else:
        print("Invalid choice!")
```

✓ **Output :**

```
┌─tushar@tushar in ~/Documents/FP/9 via  v3.10.10 took 18ms
└─λ python second.py
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 1
Enter the item to be pushed: 1
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 1
Enter the item to be pushed: 2
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 1
Enter the item to be pushed: 3
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 1
Enter the item to be pushed: 4
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
```

```
Enter your choice: 4
Stack:  ['1', '2', '3', '4']
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 3
Top item:  4
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 2
Popped item:  4
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 4
Stack:  ['1', '2', '3']
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 3
Top item:  3
****************
|  1. Push     |
|  2. Pop      |
|  3. Peek     |
|  4. Display  |
|  5. Quit     |
****************
Enter your choice: 5
```