Software Engineering
Lab-5

Name: Boricha Vinal
ID: 202001062

—

Python Code

```python
class Library:
    def __init__(self, listofBooks):
        self.books = listofBooks

    def displayAvailableBooks(self)-> None:
        bookavailable
        print(f"\n{len(self.books)} AVAILABLE BOOKS ARE: ")
        for book in self.books:
            print(" ♦-- " + book)
        print("\n")

    def borrowBook(self, name, bookname):
        if bookname not in self.books:
            print(
                f"{bookname} BOOK IS NOT AVAILABLE EITHER TAKEN BY SOMEONE ELSE, WAIT UNTIL HE RETURNED.\n")
        else:
            track.append({name: bookname})
            print("BOOK ISSUED : THANK YOU KEEP IT WITH CARE AND RETURN ON TIME.\n")
            self.books.remove(bookname)

    def returnBook(self, bookname)-> None:
        returnbook
        print("BOOK RETURNED : THANK YOU! \n")
        self.books.append(bookname)

    def donateBook(self, bookname):
        print("BOOK DONATED : THANK YOU VERY MUCH, HAVE A GREAT DAY AHEAD.\n")
        self.books.append(bookname)

```

```
class Student():
    def requestBook(self):
        print("So, you want to borrow book!")
        self.book = input("Enter name of the book you want to borrow: ")
        return self.book

    def returnBook(self)-> None:
        print("So, you want to return book!")
        name = input("Enter your name: ")
        self.book = input("Enter name of the book you want to return: ")
        if {name: self.book} in track:
            track.remove({name: self.book})
        return self.book

    def donateBook(self):
        print("Okay! you want to doante book!")
        self.book = input("Enter name of the book you want to donate: ")
        return self.book


if __name__ == "__main__":

    Delhilibrary = Library(
        ["vistas", "invention", "rich&poor", "indian", "macroeconomics", "microeconomics"])
    student = Student()
    track = []

    print("\t\t\t\t\t\t\t\t♦♦♦♦♦♦♦ WELCOME TO THE DELHI LIBRARY ♦♦♦♦♦♦♦\n")
    print("""CHOOSE WHAT YOU WANT TO DO:-\n1. Listing all books\n2. Borrow books\n3. Return books\n4. Donate books\n5. Track books\n6. exit the library\n""")

```

```
68      while (True):
69          # print(track)
70          try:
71              usr_response = int(input("Enter your choice: "))
72
73              if usr_response == 1:  # listing
74                  Delhilibrary.displayAvailableBooks()
75              elif usr_response == 2:  # borrow
76                  Delhilibrary.borrowBook(
77                      input("Enter your name: "), student.requestBook())
78              elif usr_response == 3:  # return
79                  Delhilibrary.returnBook(student.returnBook())
80              elif usr_response == 4:  # donate
81                  Delhilibrary.donateBook(student.donateBook())
82              elif usr_response == 5:  # track
83                  for i in track:
84                      for key, value in i.items():
85                          holder = key
86                          book = value
87                          print(f"{book} book is taken/issued by {holder}.")
88                  print("\n")
89                  if len(track) == 0:
90                      print("NO BOOKS ARE ISSUED!. \n")
91
92              elif usr_response == 6: #exit
93                  print("THANK YOU ! \n")
94                  exit()
95              else:
96                  print("INVAILD INPUT! \n")
97          except Exception as e:              #catch errors
98              print(f"{e}---> INVALID INPUT! \n")
```

Error caught

```
PS C:\Users\student\Downloads> py -m mypy "hello.py"
hello.py:13: error: Name "bookavailable" is not defined  [name-defined]
hello.py:29: error: Name "returnbook" is not defined  [name-defined]
hello.py:50: error: No return value expected  [return-value]
hello.py:63: error: Need type annotation for "track" (hint: "track: List[<type>] = ...")  [var-annotated]
hello.py:79: error: "returnBook" of "Student" does not return a value  [func-returns-value]
Found 5 errors in 1 file (checked 1 source file)
PS C:\Users\student\Downloads>
```

Reducing the number of errors

```
PS C:\Users\student\Downloads> py -m mypy "hello.py"
hello.py:29: error: Name "returnbook" is not defined  [name-defined]
hello.py:50: error: No return value expected  [return-value]
hello.py:63: error: Need type annotation for "track" (hint: "track: List[<type>] = ...")  [var-annotated]
hello.py:79: error: "returnBook" of "Student" does not return a value  [func-returns-value]
Found 4 errors in 1 file (checked 1 source file)
PS C:\Users\student\Downloads>
```

```
PS C:\Users\student\Downloads> py -m mypy "hello.py"
hello.py:50: error: No return value expected  [return-value]
hello.py:63: error: Need type annotation for "track" (hint: "track: List[<type>] = ...")  [var-annotated]
hello.py:79: error: "returnBook" of "Student" does not return a value  [func-returns-value]
Found 3 errors in 1 file (checked 1 source file)
```

Zero errors

```
PS C:\Users\student\Downloads> py -m mypy "hello.py"
Success: no issues found in 1 source file
PS C:\Users\student\Downloads>
)
```

Code with zero errors

```python
class Library:
    def __init__(self, listofBooks):
        self.books = listofBooks

    def displayAvailableBooks(self)-> None:
        # bookavailable
        print(f"\n{len(self.books)} AVAILABLE BOOKS ARE: ")
        for book in self.books:
            print(" ♦-- " + book)
        print("\n")

    def borrowBook(self, name, bookname):
        if bookname not in self.books:
            print(
                f"{bookname} BOOK IS NOT AVAILABLE EITHER TAKEN BY SOMEONE ELSE, WAIT UNTIL HE RETURNED.
        else:
            track.append({name: bookname})
            print("BOOK ISSUED : THANK YOU KEEP IT WITH CARE AND RETURN ON TIME.\n")
            self.books.remove(bookname)

    def returnBook(self, bookname)-> None:
        #returnbook
        print("BOOK RETURNED : THANK YOU! \n")
        self.books.append(bookname)

    def donateBook(self, bookname):
        print("BOOK DONATED : THANK YOU VERY MUCH, HAVE A GREAT DAY AHEAD.\n")
        self.books.append(bookname)
```

```
class Student():
    def requestBook(self):
        print("So, you want to borrow book!")
        self.book = input("Enter name of the book you want to borrow: ")
        return self.book

    def returnBook(self)-> None:
        print("So, you want to return book!")
        name = input("Enter your name: ")
        self.book = input("Enter name of the book you want to return: ")
        if {name: self.book} in track:
            track.remove({name: self.book})
        return self.book

    def donateBook(self):
        print("Okay! you want to doante book!")
        self.book = input("Enter name of the book you want to donate: ")
        return self.book


if __name__ == "__main__":

    Delhilibrary = Library(
        ["vistas", "invention", "rich&poor", "indian", "macroeconomics", "microeconomics"])
    student = Student()
    track = []

    print("\t\t\t\t\t\t\t\t♦♦♦♦♦♦♦ WELCOME TO THE DELHI LIBRARY ♦♦♦♦♦♦♦\n")
    print("""CHOOSE WHAT YOU WANT TO DO:-\n1. Listing all books\n2. Borrow books\n3. Return books\n4. Donate books\n5. Track books\n6. exit the library\n""")

```

```
68      while (True):
69          # print(track)
70          try:
71              usr_response = int(input("Enter your choice: "))
72
73              if usr_response == 1:  # listing
74                  Delhilibrary.displayAvailableBooks()
75              elif usr_response == 2:  # borrow
76                  Delhilibrary.borrowBook(
77                      input("Enter your name: "), student.requestBook())
78              elif usr_response == 3:  # return
79                  Delhilibrary.returnBook(student.returnBook())
80              elif usr_response == 4:  # donate
81                  Delhilibrary.donateBook(student.donateBook())
82              elif usr_response == 5:  # track
83                  for i in track:
84                      for key, value in i.items():
85                          holder = key
86                          book = value
87                          print(f"{book} book is taken/issued by {holder}.")
88                  print("\n")
89                  if len(track) == 0:
90                      print("NO BOOKS ARE ISSUED!. \n")
91
92              elif usr_response == 6: #exit
93                  print("THANK YOU ! \n")
94                  exit()
95              else:
96                  print("INVAILD INPUT! \n")
97          except Exception as e:              #catch errors
98              print(f"{e}---> INVALID INPUT! \n")
```