TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

LAB 5

By:

Sinjal Dahal (081/BEL/080)

DEPARTMENT OF COMPUTER ENGINEERING

LALITPUR, NEPAL

1. Create a class Book with attributes title, author, and price. Write a constructor to initialize these values and create an object with sample data.

- *Add a method display_info() to the Book class that prints the book's title, author, and price. Call this method using a Book object.*
- *Add a method update_price(new_price) to the Book class that updates the book's price. Demonstrate how to use it with an object.*

Code:

```
class Book:
    def __init__(self,title,author,price):
        self.title = title
        self.author = author
        self.price = price

    def display_info(self):
        print("Title \t\t Author \t\t price \n ")
        print(f"{self.title} \t\t {self.author} \t\t {self.price}\n")

    def update_price(self,new_price):
        self.price = new_price

book1 = Book("Maths","dr Dahal",1069)
book1.display_info()
```

```
book1.update_price(500)
book1.display_info()
```

Output:

```
Title           Author          price
Maths           dr Dahal        1069
Title           Author          price
Maths           dr Dahal        500
```

2. Create a class Student with attributes name and marks. Create three objects of the class and display their details using a method show_details().

Code:

```
class Student:
    def __init__(self,name,marks):
        self.name = name
        self.marks = marks

    def show_details(self):
        print(f"Name : {self.name} \t\t Marks : {self.marks} \n")

roll_1 = Student("Ram",96)
roll_2 = Student("Shyam",69)
```

```
roll_3 = Student("Hari",76)


roll_1.show_details()
roll_2.show_details()
roll_3.show_details()
```

Output:

```
Name : Ram                  Marks : 96

Name : Shyam                Marks : 69

Name : Hari                 Marks : 76
```

3. Create a class BankAccount with attributes account_holder, account_number, and balance.

- *Add methods deposit(amount) and withdraw(amount) that update the balance.*
- *Add a method show_balance() that prints the current balance.*
- *Create an object and perform a deposit, a withdrawal, and show the balance.*

Code:

```
class BankAccount:
```

```
    def __init__(self, account_holder , account_number ):
        self.account_holder = account_holder
        self.account_number = account_number
        self.balance = 0

    def deposit(self,deposit):
        self.balance += deposit

    def withdraw(self,withdraw):
        if (self.balance < withdraw):
            print("Insufficient balance.....")
        else:
            self.balance -= withdraw

    def show_balance(self):
        print(f"Balance : {self.balance} \n")

user_1 = BankAccount("Durga",1 )
user_1.deposit(100000000000)
user_1.withdraw(2500000)
user_1.show_balance()
```

Output:

```
Balance : 99997500000
```