

## 9. Programming on class

### 9.1 Aim

Write a program to create a Bank account and Rectangular using class object with initial parameter for the program statement given below.

### 9.2 Software Used

1. Anaconda Navigator
2. Jupyter Notebook

### 9.3 Pre Lab Questions

1. What is a class? How to create a class
2. Explain the following term “self” and `__init__` in class

### 9.4 (a) Program statement 1

1. Create a Python class called **BankAccount** which represents a bank account, having as attributes: **accountNumber** (numeric type), **name** (name of the account owner as string type), **balance**.
2. Create a **constructor** with parameters: **accountNumber**, **name**, **balance**.
3. Create a **Deposit()** method which manages the deposit actions.
4. Create a **Withdrawal() method** which manages withdrawals actions.
5. Create an **bankFees()** method to apply the bank fees with a percentage of 5% of the balance account.
6. Create a **display()** method to display account details.
7. Give the complete code for the **BankAccount class**.

### 9.5 (a) Program 1 Code:

```
class BankAccount:
    # create the constructor with parameters: accountNumber, name and balance
    # create Deposit() method
    # create Withdrawal method
    # create bankFees() method
    # create display() method
# Testing the code :
# Creating Withdrawal Test
# Create deposit test
# Display account informations
```

### Output Response:

Account Number :

Account Name :

Account Balance :

**Code:**

```
class BankAccount:
    def __init__(self, accountNumber, name):
        self.accountNumber = accountNumber;
        self.name = name;
        self.balance = 0;

    def bankFees(self):
        self.balance = self.balance - (5/100)*self.balance;
        print("5% bank fees applied for withdrawal");
        return;

    def deposit(self, amount):
        self.balance = self.balance + amount;
        print("Current balance after deposit:", self.balance)
        return self.balance;

    def withdrawal(self, amount):
        self.balance = self.balance - amount;
        self.bankFees();
        print("Current balance after withdrawal:", self.balance)
        return self.balance;

    def details(self):
        print("Account No.:", self.accountNumber);
        print("Name:", self.name);
        print("Current Balance:", self.balance);
        return;
```

**Observation:**

```

class BankAccount:
    def __init__(self, accountNumber, name):
        self.accountNumber = accountNumber;
        self.name = name;
        self.balance = 0;

    def bankFees(self):
        self.balance = self.balance - (5/100)*self.balance;
        print("5% bank fees applied for withdrawal");
        return;

    def deposit(self, amount):
        self.balance = self.balance + amount;
        print("Current balance after deposit:", self.balance)
        return self.balance;

    def withdrawal(self, amount):
        self.balance = self.balance - amount;
        self.bankFees();
        print("Current balance after withdrawal:", self.balance)
        return self.balance;

    def details(self):
        print("Account No.:", self.accountNumber);
        print("Name:", self.name);
        print("Current Balance:", self.balance);
        return;

```

### Output:

```

: person = BankAccount(123456, "Kunal Keshan")
  person.deposit(10000)
  person.withdrawal(500)
  person.details()

```

```

Current balance after deposit: 10000
5% bank fees applied for withdrawal
Current balance after withdrawal: 9025.0
Account No.: 123456
Name: Kunal Keshan
Current Balance: 9025.0

```

### 9.4(b) Program statement 2

1. Write a **Rectangle** class in Python language, allowing you to build a rectangle with **length** and **width** attributes.
2. Create a **Perimeter()** method to calculate the perimeter of the rectangle and a **Area()** method to calculate the area of the rectangle.
3. Create a method **display()** that display the length, width, perimeter and area of an object created using an instantiation on rectangle class.
4. Create a **Parallelepiped** child class **inheriting** from the **Rectangle** class and with a **height** attribute and another **Volume()** method to calculate the volume of the **Parallelepiped**.

### 9.5(b) Program 2 Code

class Rectangle:

```

    # define constructor with attributes: length and width
    # Create Perimeter method
    # Create area method

```

```
# create display method
# define Volume method
```

**output response:**

The length of rectangle is: 7

The width of rectangle is: 5

The perimeter of rectangle is: 24

The area of rectangle is: 35

-----

the volume of myParallelepiped is: 70

**Code:**

```
class Rectangle:
```

```
    def __init__(self, length, width):
```

```
        self.length = length;
```

```
        self.width = width;
```

```
    def perimeter(self):
```

```
        return 2*(self.length+self.length)
```

```
    def area(self):
```

```
        return self.length*self.width;
```

```
    def details(self):
```

```
        self.perimeter();
```

```
        self.area();
```

```
        print("Length:", self.length);
```

```
        print("Width:", self.width);
```

```
        print("Perimeter:", self.perimeter());
```

```
        print("Area:", self.area());
```

```
class Parallelepiped(Rectangle):
```

```
    def __init__(self, length, width, height):
```

```
        super().__init__(length, width)
```

```
        self.height = height;
```

```
    def volume(self):
```

```
        volume = self.length*self.height*self.width;
```

```
        print("Volume is:", volume)
```

```
        return volume;
```

```
twod = Rectangle(6,9);
```

```
twod.perimeter();
twod.area();
twod.details();
```

```
threed = Parallelepiped(6,9,9)
threed.volume()
```

### Observation:

```
class Rectangle:
    def __init__(self, length, width):
        self.length = length;
        self.width = width;

    def perimeter(self):
        return 2*(self.length+self.width)

    def area(self):
        return self.length*self.width;

    def details(self):
        self.perimeter();
        self.area();
        print("Length:", self.length);
        print("Width:", self.width);
        print("Perimeter:", self.perimeter());
        print("Area:", self.area());

class Parallelepiped(Rectangle):
    def __init__(self, length, width, height):
        super().__init__(length, width)
        self.height = height;

    def volume(self):
        volume = self.length*self.height*self.width;
        print("Volume is:", volume)
        return volume;

twod = Rectangle(6,9);
twod.perimeter();
twod.area();
twod.details();

threed = Parallelepiped(6,9,9)
threed.volume()
```

### Output:

```
twod = Rectangle(6,9);
twod.perimeter();
twod.area();
twod.details();

threed = Parallelepiped(6,9,9)
threed.volume()

Length: 6
Width: 9
Perimeter: 24
Area: 54
Volume is: 486

In [ ]: 486
```

## 9.6 Post Lab Questions

1. Create a Circle class and initialize it with radius. Make two methods getArea and get Circumference inside this class.
2. Create a Temperature class. Make two methods :
  1. convertFahrenheit - It will take celsius and will print it into Fahrenheit.
  2. convertCelsius - It will take Fahrenheit and will convert it into Celsius.

## 9.7 Result

Write a program to create a Bank account and Rectangular using class object with initial parameter for the program statement