PYTHON PROGRAMMING

SLOT 6

1. Write a Python class which has two methods get_String and print_String. get_String accept a string from the user and print_String print the string in upper case.

PROGRAM

```
class getPrint:
    def __init__(self):
        self.str = str
    def getString(self):
        self.str = input('Enter the string : ')
    def printString(self):
        print('Entred string is : ' , self.str.upper())

str = getPrint()
str.getString()
str.printString()
```

OUTPUT

```
Enter the string : marthoma college ayur
Entred string is :  MARTHOMA COLLEGE AYUR
```

2. Write a **Rectangle class** in Python language, allowing you to build a rectangle with **length** and **width** attributes.

PROGRAM

```
class Rectange:
    def __init__(self , length , breadth):
        self.length = length
        self.breadth = breadth

    def area(self):
        area = self.length * self.breadth
        print(area)

length = int(input('Enter the length : '))
breadth = int(input('Enter the breadth : '))
rect = Rectange(length , breadth)
rect.area()
```

OUTPUT

```
Enter the length : 5
Enter the breadth : 6
Area of rectangle is :  30
```

3. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

PROGRAM

```
class Circle:
    def __init__(self , radius):
        self.r = radius

    def area(self):
        area = 3.14 * self.r * self.r
        print('Area is ' , area)

    def per(self):
        per = 2 * 3.14 * self.r
        print('Perimeter is ' , per)

radius = int(input('Enter the radius : '))
obj = Circle(radius)
obj.area()
obj.per()
```

OUTPUT

```
Enter the radius : 10
Area is  314.0
Perimeter is  62.800000000000004
```

4. Define a **Book class** with the following attributes: **Title, Author** (Full name), **Price**.
Define a **constructor** used to initialize the attributes of the method with values entered by the user.
Set the **View() method** to display information for the current book.

PROGRAM

```
class Book:
    def __init__(self, Title, Author, Price):
        self.Title = Title
        self.Author = Author
        self.Price = Price

    def view(self):
        print('Current Book Details')
        print('Title : ' , self.Title)
        print('Author: ' , self.Author)
        print('Price : ' , self.Price)

title = input('Enter the title : ')
author = input('Enter the author name : ')
price = input('Enter the price of book : ')
obj = Book(title , author, price)
obj.view()
```

OUTPUT

Enter the title : Wings Of Fire

Enter the author name : APJ Abdhul Kalam

Enter the price of book : 300

Book Details

Title : Wings Of Fire

Author: APJ Abdhul Kalam

Price : 300

5. 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.
Create a **constructor** with parameters: **accountNumber, name, balance**.
Create a **Deposit()** method which manages the deposit actions.
Create a **Withdrawal() method** which manages withdrawals actions.
Create an **bankFees()** method to apply the bank fees with a percentage of 5% of the balance account.
Create a **display()** method to display account details.

PROGRAM

```
class BankAccount:
    def __init__(self , Account_Number , Name , Balance):
        self.Account_Number  = Account_Number
        self.Name = Name
        self.Balance = Balance

    def deposit(self , d_amnt):
        self.Balance = self.Balance + d_amnt

    def withdraw(self, w_amnt):
        if(self.Balance < w_amnt):
            print('No enough balance')
        else:
            self.Balance = self.Balance - w_amnt

    def bankFee(self):
        self.Balance = self.Balance - (self.Balance * .05)
```

```
    def disp(self):
        print('*******Details******')
        print('Account number : ' , self.Account_Number)
        print('Name : ', self.Name)
        print('Balance : ' ,self.Balance)

Account_Number = int(input('Enter the account number : '))
Name = input('Enter the name of customer : ')
Balance = int(input('Enter the current balanace : '))
deposit = int(input('Enter the deposit amount : '))
withdraw = int(input('Enter the withdraw amount : '))
obj = BankAccount(Account_Number , Name , Balance)
obj.deposit(deposit)
obj.withdraw(withdraw)
obj.bankFee()
obj.disp()
```

OUTPUT

```
Enter the account number : 7000200
Enter the name of customer : Aby
Enter the current balanace : 25000
Enter the deposit amount : 50000
Enter the withdraw amount : 1200
*******Details******
Account number :  7000200
Name :  Aby
Balance :  70110.0
```

6. Create a **Bus** child class that inherits from the Vehicle class. The default fare charge of any vehicle is **seating capacity * 100**. If Vehicle is **Bus** instance, we need to add an extra 10% on full fare as a maintenance charge. So total fare for bus instance will become the **final amount = total fare + 10% of the total fare.**

Note: The bus seating capacity is **50**. so the final fare amount should be **5500.** You need to override the fare() method of a Vehicle class in Bus class.

PROGRAM

```
class Vehicle:
    def __init__(self,name,seat_cap,milage):
        self.seat_cap = seat_cap
        self.name = name
        self.milage = milage

    def fare(self):
        return self.seat_cap * 100

class Bus(Vehicle):
    def fare(self):
        bus_fare = (super().fare() + super().fare() * .1)
        return bus_fare

name = (input("Enter the vehicle name : "))
seat = int(input("Enter the capacity : "))
milage = int(input("Enter Mileage : "))
print('********************')
print('Vechicle name : ' , name)
```

```
print('Seating capacity of vechicle : ', seat)
print('Milage of vechicle : ' , milage)
bus = Bus(name , seat , milage)
print("Fare : ",bus.fare())
```

OUTPUT

```
Enter the vehicle name : Bus
Enter the capacity : 45
Enter Mileage : 10
********************
Vechicle name :  Bus
Seating capacity of vechicle :  45
Milage of vechicle :  10
Fare :  4950.0
```

Searching and Sorting

7. Implement Linear Search.

PROGARAM

```
class Linear:
    def __init__(self , list):
        self.list = list

    def search(self , key):
        count = 0
        for i in range(0 , len(list)-1):
            if key == list[i]:
                count  += 1
                break
        if count != 0:
            print('Element is present at ' , i + 1)
        else:
            print('Element is not present ')

list = list(map(int,input('Enter list elements : ').split()))
key = int(input('Enter the element for search : '))
obj = Linear(list)
obj.search(key)
```

OUTPUT

Enter list elements : 56 34 90 54 12 27 98 77 74 66

Enter the element for search : 54

Element is present at 4

8. Implement Binary Search

PROGRAM

```
class Binary:
    def search(self ,list ,key):
        low = 0
        high = len(list) – 1

        while(low <= high):
            mid = (low + high)//2
            if(list[mid] == key):
                print('Element present at index ', mid + 1)
                break
            elif(key < list[mid]):
                high = mid-1
            elif(key > list[mid]):
                low = mid + 1

        if(low > high):
            print('Element not present')

list = list(map(int,input('Enter the sorted elements : ').split()))
key = int(input('Enter the search element : '))
obj = Binary()
obj.search(list , key)
```

OUTPUT

Enter the sorted elements : 78 99 102 234 564 777

Enter the search element : 102

Element present at index 3

9. Implement Bubble sort

PROGRAM

```
class Bubble:
    def sort(self , list):
        for i in range (0,len(list)):
            for j in range (1,len(list)-i):
                if(list[j]  <  list[j-1]):
                    list[j] , list[j-1]  =  list[j-1] , list[j]
        print('Sorted array  : ', list)

list = list(map(int,input('Enter array elements : ').split()))
print('Orginal Array : ', list)
obj = Bubble()
obj.sort(list)
```

OUTPUT

```
Enter array elements : 32 12 56 5 3 22
Orginal Array :  [32, 12, 56, 5, 3, 22]
Sorted array  :  [3, 5, 12, 22, 32, 56]
```

10. Implement insertion sort

PROGRAM

```
class Insertion:
    def sort(self , array):
        for i in range (0,len(array)):
            j = i
            while j > 0 and array[j] < array[j-1]:
                temp = array[j]
                array[j] = array[j-1]
                array[j-1] = temp
                j = j-1
        print('Sorted array  : ' , array)

array = list(map(int,input('Enter the array elements : ').split()))
print('Orginal array : ', array)
obj = Insertion()
obj.sort(array)
```

OUTPUT

```
Enter the array elements : 67 45 23 98 80 50 34
Orginal array :  [67, 45, 23, 98, 80, 50, 34]
Sorted array  :  [23, 34, 45, 50, 67, 80, 98]
```

11. Implement Selection Sort

PROGRAM

```
class Selection:
    def sort(self , arr):
        for i in range (0,len(arr)):
            min = i
            for j in range (i+1 , len(arr)):
                if arr[j] < arr[min]:
                    min = j
            arr[i] , arr[min] = arr[min] , arr[i]
        print('Sorted array : ', arr)

arr = list(map(int,input('Enter the array elements : ').split()))
print('Orginal array : ' , arr)
obj = Selection()
obj.sort(arr)
```

OUTPUT

```
Enter the array elements : 90 45 55 34 23 78 100
Orginal array :  [90, 45, 55, 34, 23, 78, 100]
Sorted array :  [23, 34, 45, 55, 78, 90, 100]
```