Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Python Programming - 2301CS404

Lab - 13

Smit Maru - 23010101161 - 260

OOP

01) Write a Program to create a class by name Students, and initialize attributes like name, age, and grade while creating an object.

In [7]:
```
class Students:
    def __init__(self,name,age,grade):
        self.name = name
        self.age = age
        self.grade = grade

stu = Students("Smit",20,"A+")

print("Name =",stu.name)
print("Age =",stu.age)
print("Grade =",stu.grade)
```

```
Name = Smit
Age = 20
Grade = A+
```

02) Create a class named Bank_Account with Account_No, User_Name, Email,Account_Type and Account_Balance data members. Also create a method GetAccountDetails() and DisplayAccountDetails(). Create main method to demonstrate the Bank_Account class.

In [11]:
```
class Bank_Account:
    def __init__(self, Account_No, User_Name, Email, Account_Type, Account_Balance)
        self.Account_No = Account_No
        self.User_Name = User_Name
        self.Email = Email
        self.Account_Type = Account_Type
        self.Account_Balance = Account_Balance

    def GetAccountDetails(self):
        return {
            "Account_No" : self.Account_No,
            "User_Name" : self.User_Name,
            "Email" : self.Email,
            "Account_Type" : self.Account_Type,
            "Account_Balance" : self.Account_Balance
        }

    def DisplayAccountDetails(self):
        Account_Details = self.GetAccountDetails()
        for key, value in Account_Details.items():
            print(f"{key} : {value}")

Account = Bank_Account(123456789, "Smit", "smit@gmail.com", "Current",100000000)
Account.DisplayAccountDetails()
```

```
Account_No : 123456789
User_Name : Smit
Email : smit@gmail.com
Account_Type : Current
Account_Balance : 100000000
```

03) WAP to create Circle class with area and perimeter function to find area and perimeter of circle.

In [ ]:
```
import math

class circle:
    def __init__(self,radius):
        self.radius = radius

    def area(self):
        return math.pi * (self.radius ** 2)

    def perimeter(self):
        return 2 * math.pi * self.radius

circle = Circle(1)
print("Area :",circle.area())
print("Perimeter :",circle.perimeter())
```

```
Area : 3.141592653589793
Perimeter : 6.283185307179586
```

04) Create a class for employees that includes attributes such as name, age, salary, and methods to update and display employee information.

In [23]:
```
class Employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary

    def update_salary(self,new_salary):
        self.salary = new_salary

    def update_age(self, new_age):
        self.age = new_age

    def display_information(self):
        print("Name =",self.name)
        print("Age =",self.age)
        print("Salary =",self.salary)

employee = Employee("Smit",20, 100000)
employee.display_information()

employee.update_salary(150000)
employee.update_age(22)
employee.display_information()
```

```
Name = Smit
Age = 20
Salary = 100000
Name = Smit
Age = 22
Salary = 150000
```

05) Create a bank account class with methods to deposit, withdraw, and check balance.

In [26]:
```
class BankAccount:
    def __init__(self, initial_balance=0):
        self.balance = initial_balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited {amount}. New balance is {self.balance}.")
        else:
            print("Enter Positive Deposit Amount.")

    def withdraw(self, amount):
        if amount > 0 and amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew {amount}. New balance is {self.balance}.")
```

```
        else:
            print("Invalid withdrawal amount or insufficient balance.")

    def check_balance(self):
        print(f"Your current balance is {self.balance}.")
```

06) Create a class for managing inventory that includes attributes such as item name, price, quantity, and methods to add, remove, and update items.

In [ ]:
```
class InventoryItem:
    def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

    def add(self, amount):
        self.quantity += amount

    def remove(self, amount):
        if amount <= self.quantity:
            self.quantity -= amount
        else:
            print("Not enough items to remove.")

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

    def __str__(self):
        return f"Item: {self.name}, Price: {self.price}, Quantity: {self.quantity}"
```

07) Create a Class with instance attributes of your choice.

In [ ]:
```
class Dog:
    def __init__(self, name, breed, age):
        self.name = name
        self.breed = breed
        self.age = age

my_dog = Dog("Lucky", "Golden Retriever", 3)

print(my_dog.name)
print(my_dog.breed)
print(my_dog.age)
```

```
Lucky
Golden Retriever
3
```

08) Create one class student_kit

Within the student_kit class create one class attribute principal name (Mr ABC)

Create one attendance method and take input as number of days.

While creating student take input their name .

Create one certificate for each student by taking input of number of days present in class.

In [ ]:
```
class student_kit:
    principal_name = "Mr ABC"

    def __init__(self, student_name):
        self.student_name = student_name
        self.days_present = 0

    def attendance(self, days):
        self.days_present = days

    def create_certificate(self):
        return f"Certificate of {self.student_name}: Present for {self.days_present
```

09) Define Time class with hour and minute as data member. Also define addition method to add two time objects.

In [33]:
```
class Time:
    def __init__(self, hour, minute):
        self.hour = hour
        self.minute = minute

    def add(self, other):
        total_minutes = self.minute + other.minute
        total_hours = self.hour + other.hour + total_minutes // 60
        total_minutes = total_minutes % 60
        return Time(total_hours, total_minutes)


time1 = Time(1, 30)
time2 = Time(2, 45)
result = time1.add(time2)
print(f"Total Time: {result.hour} hours and {result.minute} minutes")
```

```
Total Time: 4 hours and 15 minutes
```