

# PYTHON IN 10 VIDEOS

Assignment - 10 (The End)

1. Create a class named `Student` with attributes `name` and `age`. Use a constructor to initialize these attributes. Create an object of the class and print its attributes.

2. Create an abstract base class named **Shape** with an abstract method **area**. Create derived classes **Circle** and **Square** that implement the **area** method. Create objects of the derived classes and call the **area** method.

3. Create a class named `Rectangle` with private attributes `length` and `width`. Use properties to get and set these attributes. Create an object of the class and test the properties.

4. Create a base class named `Person` with attributes `name` and `age`. Create a derived class named `Employee` that inherits from `Person` and adds an attribute `employee_id`. Create an object of the derived class and print its attributes.

5. Create a class named `BankAccount` with private attributes `account_number` and `balance`. Add methods to deposit and withdraw money, and to check the balance. Create an object of the class and perform some operations.

6. Create a base class named `Walker` with a method `walk` that prints a walking message. Create another base class named `Runner` with a method `run` that prints a running message. Create a derived class named `Athlete` that inherits from both `Walker` and `Runner`. Create an object of the `Athlete` class and call both methods.

7. Create an abstract base class named `Worker` with an abstract method `work`. Create two derived classes `Engineer` and `Doctor` that implement the `work` method. Create another derived class `Scientist` that inherits from both `Engineer` and `Doctor`. Create an object of the `Scientist` class and call the `work` method.



8. Create an abstract base class named `Appliance` with an abstract property `power`. Create two derived classes `WashingMachine` and `Refrigerator` that implement the `power` property. Create objects of the derived classes and access the `power` property.

9. Create a custom exception named `InsufficientBalanceError`. In the `BankAccount` class, raise this exception when a withdrawal amount is greater than the balance. Handle the exception and print an appropriate message.

10. Create a class named **Vector** with attributes **x** and **y**.  
Overload the **+** operator to add two **Vector** objects.  
Create objects of the class and test the operator  
overloading