


 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python	
Experiment No: 14	Date: 20/10/2025	Enrollment No: 92510133049

Aim: Practical based on OOP concept using Python

IDE:

Object Oriented Programming is a fundamental concept in Python, empowering developers to build modular, maintainable, and scalable applications. By understanding the core OOP principles classes, objects, inheritance, encapsulation, polymorphism, and abstraction programmers can leverage the full potential of Python's OOP capabilities to design elegant and efficient solutions to complex problems.



 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python	
Experiment No: 14	Date: 20/10/2025	Enrollment No: 92510133049

OOPs Concepts in Python

- Class in Python
- Objects in Python
- Polymorphism in Python
- Encapsulation in Python
- Inheritance in Python
- Data Abstraction in Python

Python Class

A class is a collection of objects. A class contains the blueprints or the prototype from which the objects are being created. It is a logical entity that contains some attributes and methods.

Defining a Class

Example 1:

class Car:

Constructor to initialize the object

```
def __init__(self, brand, model):
    self.brand = brand # Attribute
    self.model = model # Attribute
```


Method to describe the car

```
def car_details(self):
    return f"Car: {self.brand}, Model: {self.model}"
```

Creating an object of the Car class

```
my_car = Car("Toyota", "Corolla")
print(my_car.car_details())
```

Output:

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python	
Experiment No: 14	Date: 20/10/2025	Enrollment No: 92510133049

```

C: > Users > PRAVEEN KUMAR > exp - 14.py > ...
1  class Car:
2      # Constructor to initialize the object
3      def __init__(self, brand, model):
4          self.brand = brand # Attribute
5          self.model = model # Attribute
6
7      # Method to describe the car
8      def car_details(self):
9          return f"Car: {self.brand}, Model: {self.model}"
10
11 # Creating an object of the Car class
12 my_car = Car("Mahindra", "Thar")
13 print(my_car.car_details())

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + - [] [x] [y] [z] [w] [v] [u] [t] [s] [r] [q] [p] [o] [n] [m] [l] [k] [j] [i] [h] [g] [f] [e] [d] [c] [b] [a]

```

PS C:\Users\PRAVEEN KUMAR> & "C:/Users/PRAVEEN KUMAR/AppData/Local/Microsoft/WindowsApps/python3.13.exe" "c:/Users/PRAVEEN KUMAR/exp - 14.py"
Car: Mahindra, Model: Thar
PS C:\Users\PRAVEEN KUMAR>

```

Example 2:

Class with Methods and Attributes

class Rectangle:

```

def __init__(self, width, height):
    self.width = width
    self.height = height

```

Method to calculate area

```

def area(self):
    return self.width * self.height

```

Method to calculate perimeter

```

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



```

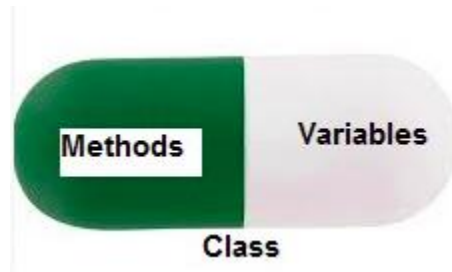
Create an object

```

rect = Rectangle(10, 5)

```


 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python	
Experiment No: 14	Date: 20/10/2025	Enrollment No: 92510133049



Example 3:

class BankAccount:

```
def __init__(self, account_holder, balance):
    self.account_holder = account_holder
    self.__balance = balance # Private attribute
```

```
def deposit(self, amount):
    self.__balance += amount
```



```
def withdraw(self, amount):
    if amount <= self.__balance:
        self.__balance -= amount
    else:
        print("Insufficient funds")
```

```
def get_balance(self):
    return self.__balance
```

Create an account

```
account = BankAccount("John", 1000)
account.deposit(500)
print(account.get_balance()) #
account.withdraw(700)
print(account.get_balance()) #
```

Output

 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python	
Experiment No: 14	Date: 20/10/2025	Enrollment No: 92510133049



Dog class inherits from Animal class

```
class Dog(Animal):
    def speak(self):
        return f"{self.name} says Woof!"
```

Cat class inherits from Animal class

```
class Cat(Animal):
    def speak(self):
        return f"{self.name} says Meow!"
```

```
dog = Dog("Buddy")
cat = Cat("Whiskers")
print(dog.speak()) #
print(cat.speak()) #
Output :
```


 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python	
Experiment No: 14	Date: 20/10/2025	Enrollment No: 92510133049

```
class Square(Polygon):
    # renders Square
    def render(self):
        print("Rendering Square...")
```

```
class Circle(Polygon):
    # renders circle
    def render(self):
        print("Rendering Circle...")
```


```
# create an object of Square
s1 = Square()
s1.render()
```

```
# create an object of Circle
c1 = Circle()
c1.render()
```

Output:

**Enrollment No: 92510133049**

```
PS C:\Users\PRAVEEN KUMAR>
```


 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python
Experiment No: 14	Date: 20/10/2025 Enrollment No: 92510133049

ANSWERS:

Write a Python program to create a class representing a Circle. Include methods to calculate its area and perimeter.

```

C: > Users > PRAVEEN KUMAR > exp - 14.py > ...
1  import math
2
3  class Circle:
4      def __init__(self, radius):
5          self.radius = radius
6
7      def area(self):
8          """Calculate and return the area of the circle."""
9          return math.pi * self.radius ** 2
10
11     def perimeter(self):
12         """Calculate and return the perimeter (circumference) of the circle."""
13         return 2 * math.pi * self.radius
14
15     # Example usage
16     if __name__ == "__main__":
17         r = float(input("Enter the radius of the circle: "))
18         c = Circle(r)
19         print(f"Area of the circle: {c.area():.2f}")
20         print(f"Perimeter of the circle: {c.perimeter():.2f}")



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - [] [X] ... [] [X]

```

PS C:\Users\PRAVEEN KUMAR> & "C:/Users/PRAVEEN KUMAR/AppData/Local/Microsoft/WindowsApps/python3.13.exe" "c:/Users/PRAVEEN KUMAR/exp - 14.py"
Enter the radius of the circle: 3.4
Area of the circle: 36.32
Perimeter of the circle: 21.36
PS C:\Users\PRAVEEN KUMAR>

```

 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Engineering & Technology Department of Information and Communication Technology	
Subject: Programming With Python (01CT1309)	Aim: Practical based on OOP concept using Python	
Experiment No: 14	Date: 20/10/2025	Enrollment No: 92510133049

- Create a class `Book` that stores details like the title, author, and price of a book. Add methods to display the details of the book and apply a discount to the price. (a) Create two objects for different books and display their details. (b) Apply a 10% discount to one of the books and display the updated price.

```

C: > Users > PRAVEEN KUMAR > exp - 14.py > ...
1  class Book:
2      def __init__(self, title, author, price):
3          self.title = title
4          self.author = author
5          self.price = price
6
7      def display_details(self):
8          """Display the book's details."""
9          print(f"Title: {self.title}")
10         print(f"Author: {self.author}")
11         print(f"Price: ₹{self.price:.2f}")
12
13     def apply_discount(self, discount_percent):
14         """Apply a discount to the book's price."""
15         discount_amount = self.price * (discount_percent / 100)
16         self.price -= discount_amount
17         print(f"Discount of {discount_percent}% applied. New price: ₹{self.price:.2f}")
18
19 # (a) Create two book objects and display their details
20 book1 = Book("The Alchemist", "Paulo Coelho", 499.00)
21 book2 = Book("Sapiens", "Yuval Noah Harari", 799.00)
22
23 print("■ Book 1 Details:")
24 book1.display_details()
25 print("\n■ Book 2 Details:")
26 book2.display_details()
27
28 # (b) Apply a 10% discount to book2 and display updated price
29 print("\n👉 Applying 10% discount to Book 2:")
30 book2.apply_discount(10)

```

```

PS C:\Users\PRAVEEN KUMAR> & "C:/Users/PRAVEEN KUMAR/AppData/Local/Microsoft/WindowsApps/python3.13.exe" "c:/Users/PRAVEEN KUMAR/exp - 14.py"
Title: The Alchemist
Author: Paulo Coelho
Price: ₹499.00

■ Book 2 Details:
Title: Sapiens
Author: Yuval Noah Harari
Price: ₹799.00

👉 Applying 10% discount to Book 2:
Discount of 10% applied. New price: ₹719.10
PS C:\Users\PRAVEEN KUMAR>

```