

✓ Day 7 of Training at Ansh Info Tech

Topics Covered

Object Oriented Programming in Python

- **Classes in Python**
 - **Objects in Python**
 - **Object Attributes in Python**
 - **20 Practice Questions on Python Classes and Objects**
 - **5+ Extra Practice Questions**
-

Summary

Classes in Python

Classes are blueprints for creating objects. They encapsulate data for the object and methods to manipulate that data. Classes in Python are defined using the `class` keyword.

Objects in Python

Objects are instances of classes. They represent the concrete implementation of the class blueprint and can have attributes (data) and methods (functions) defined in the class.

Object Attributes in Python

Object attributes are variables that hold data associated with a particular object. These can be defined in the class and can be accessed and modified using dot notation.

Practice Questions on Python Classes and Objects

1. Define a class named `Car` with attributes `make`, `model`, and `year`.
2. Create an instance of the `Car` class and print its attributes.
3. Add a method to the `Car` class that returns a formatted string describing the car.
4. **And many more...**

Extra Practice Questions

1. Create a class `Student` with attributes `name` and `grade`.
2. Define methods in the `Student` class to set and get the `grade`.
3. **And many more...**

Practice Questions:

1. Define a simple Car class with attributes for make, model, and year. Create an object of this class and print its attributes.
2. Create a Person class with attributes for name and age. Add a method to print a greeting message. Create an object and call the method.
3. Define a Rectangle class with attributes for length and width. Add a method to calculate the area of the rectangle. Create an object and print the area.
4. Create a Student class with attributes for name and grades (a list of numbers). Add a method to calculate the average grade. Create an object and print the average grade.
5. Define a Book class with attributes for title, author, and pages. Add a method to display the book's details. Create an object and call the method.
6. Create a Dog class with attributes for name and breed. Add a method to make the dog bark (print a bark message). Create an object and call the method.
7. Define a BankAccount class with attributes for account number and balance. Add methods to deposit and withdraw money. Create an object and test the methods.
8. Create a Circle class with attributes for radius. Add a method to calculate the circumference. Create an object and print the circumference.
9. Define a Laptop class with attributes for brand and price. Add a method to apply a discount to the price. Create an object and test the method.
10. Create a Employee class with attributes for name and salary. Add a method to give a raise (increase the salary). Create an object and test the method.
11. Define a Point class with attributes for x and y coordinates. Add a method to calculate the distance from another point. Create two objects and print the distance between them.
12. Create a Movie class with attributes for title, director, and release year. Add a method to display the movie's information. Create an object and call the method.
13. Define a Product class with attributes for name and price. Add a method to calculate the price after tax (assume a fixed tax rate). Create an object and print the price after tax.
14. Create a Player class with attributes for name and score. Add a method to update the score. Create an object and test the method.
15. Define a House class with attributes for address and number of rooms. Add a method to display the house's details. Create an object and call the method.

16. Create a Shape class with attributes for color and filled (a boolean). Add a method to display the shape's properties. Create an object and call the method.
17. Define a Vehicle class with attributes for type and speed. Add a method to accelerate (increase the speed). Create an object and test the method.
18. Create a Country class with attributes for name and population. Add a method to display the country's information. Create an object and call the method.
19. Define a Student class with attributes for name and a dictionary of subjects and grades. Add a method to calculate the average grade for all subjects. Create an object and print the average grade.
20. Create a Calendar class with attributes for day, month, and year. Add a method to display the date in DD/MM/YYYY format. Create an object and call the method.

#Question 1

```
class Car:
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def print_details(self):
        print("Make:", self.make)
        print("Model:", self.model)
        print("Year:", self.year)

car1 = Car("Toyota", "Camry", 2022)
print(car1.make)
print(car1.model)
print(car1.year)
car1.print_details()
```

#Question 2

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def greet(self):
        print(f"Hello, {self.name}. Oh, You are {self.age} Years Old.")

p1 = Person("Rahul Sachdeva", 19)
p1.greet()
```

#Question 3

```
class Rectangle:
    length = 10
    width = 20
    def area(self):
        print("The Area of the Rectangle is: "+str(self.length * self.width))

r1 = Rectangle()
r1.area()
```

#Question 4

```
class Student:
    def __init__(self, name, grades):
        self.name = name
        self.grades = grades

    def average_grade(self):
        total_grades = sum(self.grades)
```



```
average = total_grades/len(self.grades)
print("Average Grade:", average)
```

```
s1 = Student("Rahul Sachdeva", [90, 85, 78])
s1.average_grade()
```

#Question 5

```
class Book:
    def __init__(self, book, author, price):
        self.book = book
        self.author = author
        self.price = price

    def display(self):
        print("Book:", self.book)
        print("Author:", self.author)
        print("Price:", self.price)
```

```
b1 = Book("The Great Gatsby", "F. Scott Fitzgerald", 19.99)
b1.display()
```

#Question 6

```
class Dog:
    def __init__(self, name, breed):
        self.name = name
        self.breed = breed

    def bark(self):
        print("Woof!")
```

```
d1 = Dog("Buddy", "Golden Retriever")
d1.bark()
```

Question 7

```
class BankAccount:
    def __init__(self, account_number, balance):
        self.account_number = account_number
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount
        print(f"Deposited {amount}. New balance is {self.balance}.")

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



```
account1 = BankAccount("123456789", 1000)
account1.deposit(500)
account1.withdraw(300)
account1.withdraw(1500)
```

Question 8

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

    def calculate_circumference(self):
        return 2 * 3.14159 * self.radius

circle1 = Circle(5)
print(f"The circumference of the circle is {circle1.calculate_circumference()}")
```

Question 9

```
class Laptop:
    def __init__(self, brand, price):
        self.brand = brand
        self.price = price

    def apply_discount(self, discount_percentage):
        discount_amount = self.price * (discount_percentage / 100)
        self.price -= discount_amount
        print(f"Discount applied. New price is {self.price}.")

laptop1 = Laptop("Dell", 1000)
laptop1.apply_discount(10)
```

Question 10

```
class Employee:
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary

    def give_raise(self, raise_amount):
        self.salary += raise_amount
        print(f"Raise given. New salary is {self.salary}.")

employee1 = Employee("Alice", 50000)
employee1.give_raise(5000)
```

Question 11

```
import math

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
```



```

def calculate_distance(self, other_point):
    return math.sqrt((self.x - other_point.x) ** 2 + (self.y - other_point.y) ** 2)

point1 = Point(0, 0)
point2 = Point(3, 4)
print(f"The distance between the points is {point1.calculate_distance(point2)}.")

```

Question 12

```

class Movie:
    def __init__(self, title, director, release_year):
        self.title = title
        self.director = director
        self.release_year = release_year

    def display_info(self):
        print(f"Title: {self.title}")
        print(f"Director: {self.director}")
        print(f"Release Year: {self.release_year}")

```

```

movie1 = Movie("Inception", "Christopher Nolan", 2010)
movie1.display_info()

```

Question 13

```

class Product:
    def __init__(self, name, price):
        self.name = name
        self.price = price

    def price_after_tax(self, tax_rate):
        return self.price * (1 + tax_rate / 100)

```

```

product1 = Product("Laptop", 1000)
print(f"The price after tax is {product1.price_after_tax(5)}.")

```

Question 14

```

class Player:
    def __init__(self, name, score):
        self.name = name
        self.score = score

    def update_score(self, points):
        self.score += points
        print(f"Score updated. New score is {self.score}.")

```

```

player1 = Player("John", 50)
player1.update_score(10)

```

Question 15

```

class House:
    def __init__(self, address, number_of_rooms):

```



```
        self.address = address
        self.number_of_rooms = number_of_rooms

    def display_details(self):
        print(f"Address: {self.address}")
        print(f"Number of rooms: {self.number_of_rooms}")

house1 = House("123 Main St", 4)
house1.display_details()
```

Question 16

```
class Shape:
    def __init__(self, color, filled):
        self.color = color
        self.filled = filled

    def display_properties(self):
        print(f"Color: {self.color}")
        print(f"Filled: {self.filled}")
```

```
shape1 = Shape("Red", True)
shape1.display_properties()
```

Question 17

```
class Vehicle:
    def __init__(self, type, speed):
        self.type = type
        self.speed = speed

    def accelerate(self, increase):
        self.speed += increase
        print(f"Accelerated. New speed is {self.speed}.")
```

```
vehicle1 = Vehicle("Car", 60)
vehicle1.accelerate(20)
```

Question 18

```
class Country:
    def __init__(self, name, population):
        self.name = name
        self.population = population

    def display_info(self):
        print(f"Name: {self.name}")
        print(f"Population: {self.population}")
```

```
country1 = Country("Canada", 38000000)
country1.display_info()
```

Question 19

```
class Student:
```

```

def __init__(self, name, subjects_grades):
    self.name = name
    self.subjects_grades = subjects_grades

def calculate_average_grade(self):
    total_grades = sum(self.subjects_grades.values())
    average = total_grades / len(self.subjects_grades)
    return average

student1 = Student("John", {"Math": 90, "English": 85, "Science": 78})
print(f"The average grade is {student1.calculate_average_grade()}")

# Question 20
class Calendar:
    def __init__(self, day, month, year):
        self.day = day
        self.month = month
        self.year = year

    def display_date(self):
        print(f"{self.day:02}/{self.month:02}/{self.year}")

calendar1 = Calendar(13, 6, 2024)
calendar1.display_date()

```



```

Toyota
Camry
2022
Make: Toyota
Model: Camry
Year: 2022
Hello, Rahul Sachdeva. Oh, You are 19 Years Old.
The Area of the Rectangle is: 200
Average Grade: 84.33333333333333
Book: The Great Gatsby
Author: F. Scott Fitzgerald
Price: 19.99
Woof!
Deposited 500. New balance is 1500.
Withdrew 300. New balance is 1200.
Insufficient balance.
The circumference of the circle is 31.4159.
Discount applied. New price is 900.0.
Raise given. New salary is 55000.
The distance between the points is 5.0.
Title: Inception
Director: Christopher Nolan
Release Year: 2010
The price after tax is 1050.0.
Score updated. New score is 60.
Address: 123 Main St
Number of rooms: 4

```

✓ Exercise on class design - Level 1

Exercise :

Jack (24 years old) and Jill (27 years old) are two employees working in a multi-national company. They draw salaries 30000 and 40000 respectively.

Identify the class name and attributes so as to represent Jack and Jill from the list given.

Options:

Jill

salary

Jack

age

__init__

Employee

name

```
class Employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary
    def display(self):
        print("Name:", self.name)
        print("Age:", self.age)
        print("Salary:", self.salary)
```

```
jack = Employee("Jack", 24, 30000)
jack.display()
jill = Employee("Jill", 27, 40000)
jill.display()
```
