

# Day=06

## Object-Oriented programming

### Creating an Object

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

    def greet(self):
        print(f"Hello, my name is {self.name} and I am {self.age} years old.")

person1=Person("Ramesh", 25)
person1.greet()
```

Hello, my name is Ramesh and I am 25 years old.

```
class car:
    no_of_wheels = 0
    mileage = 70
    no_of_seats = 4

    def moveforward (self):
        print("moving forward")

    def movebackward(self):
        print("moving backward")

    def turnleft(self):
        print("turning left")

    def turnright(self):
        print("turning right")

car1 = car()
print(car1.no_of_seats)
print(car1.mileage)
print(car1.no_of_wheels)

car2 = Car()
print(car2.no_of_seats)
```

```

print(car2.mileage)
print(car2.no_of_wheels)

car3 = car()
car3.mileage = 100
car3.no_of_wheels = 10
car3.no_of_seats = 2
print(car3.no_of_seats)
print(car3.mileage)
print(car3.no_of_wheels)

car3.moveforward()
car3.movebackward()
car3.turnleft()
car3.turnright()

4
70
0
4
70
0
2
100
10
moving forward
moving backward
turning left
turning right

class Person:
    def __init__(self, name, age): # Changed __init__ to __init__
        self.name = name
        self._age = age
        self.__salary = 50000

    def get_salary(self):
        return self.__salary

    def set_salary(self, new_salary):
        if new_salary > 0:
            self.__salary = new_salary
        else:
            print("Salary must be positive.")

    def display_info(self):
        print(f"Name: {self.name}, Age: {self._age}, Salary: {self.__salary}")

```

```

person = Person("Alice", 30)

print(person.name)

print(person._age)

# print(person.__salary)

print(person.get_salary())

person.set_salary(60000)
person.display_info()

Alice
30
50000
Name: Alice, Age: 30, Salary: 60000

class ParentClass:
    def __init__(self, name):
        self.name = name

    def name1(self):
        print(f"Hello, my name is {self.name}")

class ChildClass(ParentClass):
    def __init__(self, name, age):
        super().__init__(name)
        self.age = age

    def print_age(self):
        print(f"{self.age} years old")

child = ChildClass("Ramesh", 12)
child.name1()
child.print_age()

Hello, my name is Ramesh
12 years old

def is_palindrome(n):
    original = n
    reversed_num = 0

    while n > 0:
        digit = n % 10
        reversed_num = reversed_num * 10 + digit
        n = n // 10

    return original == reversed_num

```

```
num = int(input("Enter a number: "))

if is_palindrome(num):
    print(f"{num} is a palindrome number.")
else:
    print(f"{num} is not a palindrome number.")

Enter a number: 202
202 is a palindrome number.
```