a number is even or odd

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

if number % 2 == 0:
    print(f"{number} is even.")
else:
    print(f"{number} is odd.")

Enter a number: 10
10 is even.
```

In the if statement we are using boolean values (true or false) , if the condition is true then it prints the given number is even otherwise it prints the given number is odd

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("Honey",19)
      person1.greet()

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

```
print(car2.no_of_seats)

car3 = car()
car3.mileage = 100
car3.no_of_wheels = 10
car3.no_of_seats = 12
print(car3.mileage)
print(car3.no_of_wheels)
print(car3.no_of_seats)
print(car3.no_of_seats)
print(car3.mileage)
print(car3.no_of_wheels)

car1.moveforward()
car1.movebackward()
car1.turnleft()
car1.turnright()
```

```
4
70
0
70
0
4
100
10
12
12
100
10
moving forward
moving backward
turning left
turning right
```

encapsulation

```
class Person:
  def __init__(self, name, age):
    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}")

perso1=Person("alice",25)
perso1.display_info()
perso1.set_salary(60000)
perso1.display_info()
```

```
Name:alice,Age:25,Salary:50000
Name:alice,Age:25,Salary:60000
```

inheritance

```
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 age1(self):
      print(f"i am {self.age} years old")

child=Childclass("Honey",18)
child.name1()
child.age1()
```

```
hello,my name is Honey
i am 18 years old
```

create a class for a bank account

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

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

  def withdraw(self,amount):
```

```
    if amount<=self.balance:
      self.balance-=amount # Corrected the typo here: changed 'amoun'
to 'amount'
      print(f"withdrew {amount}.New balance:{self.balance}")
    else:
      print("insufficient balance")
account=BankAccount("12345", 0) # Provided an initial balance of 0
account.deposite(1000)
account.withdraw(500)
```

```
Deposited 1000.New balance:1000
withdrew 500.New balance:500
```