In [4]:

```
#TASK - 01

class Calculator:
    def __init__(self) -> None:
        print("Let's calculate")
    def values(self, v1,op,v2):
        self.v1,self.v2,self.op = v1,v2,op
        print(f"Value 1: {self.v1}\nOperator: {self.op}\nValue 2: {self.v2}\

    def add(self): return self.v1+self.v2
    def sub(self): return self.v1-self.v2
    def mul(self): return self.v1*self.v2
    def div(self): return self.v1/self.v2

obj = Calculator()
first_value, operator, second_value = int(input()), input().strip(), int(inp
obj.values(first_value,operator,second_value)
if operator == '+': print(obj.add())
elif operator == '-': print(obj.sub())
elif operator == '*': print(obj.mul())
elif operator == '/': print(obj.div())
```

```
Let's calculate
1
+
2
Value 1: 1
Operator: +
Value 2: 2
Result: 3
```

In [5]:

```python
#TASK - 02

class Customer:
    def __init__(self, name) -> None:
        self.name = name
        self.purc = []
    def greet(self, name=None):
        print(f"Hello, {name}!") if name else print("Hello!")
    def purchase(self, *items):
        self.purc += items
        print(f"{self.name}, you purchased {len(self.purc)} item(s): ")
        for item in self.purc:
            print(item)


customer_1 = Customer("Sam")
customer_1.greet()
customer_1.purchase("chips", "chocolate", "orange juice")
print("----------------------------")
customer_2 = Customer("David")
customer_2.greet("David")
customer_2.purchase("orange juice")
```

```
Hello!
Sam, you purchased 3 item(s):
chips
chocolate
orange juice
----------------------------
Hello, David!
David, you purchased 1 item(s):
orange juice
```

In [7]:

```
#TASK - 03

class Panda:
    def __init__(self, name, gender, age) -> None:
        self.name, self.gender, self.age = name, gender, age
    def sleep(self, hour=0):
        if   3 <= hour <=5: return f"{self.name} sleeps {hour} hours daily a
        elif 6 <= hour <=8: return f"{self.name} sleeps {hour} hours daily a
        elif 9 <= hour<= 11: return f"{self.name} sleeps {hour} hours daily
        else: return f"{self.name}'s duration is unknown thus shouldhave onl

panda1 = Panda("Kunfu","Male", 5)
panda2=Panda("Pan Pan","Female",3)
panda3=Panda("Ming Ming","Female",8)
print("{} is a {} Panda Bear who is {} years old".format(panda1.name,panda1.
print("{} is a {} Panda Bear who is {} years old".format(panda2.name,panda2.
print("{} is a {} Panda Bear who is {} years old".format(panda3.name,panda3.
print("============================")
print(panda2.sleep(10))
print(panda1.sleep(4))
print(panda3.sleep())
```

```
Kunfu is a Male Panda Bear who is 5 years old
Pan Pan is a Female Panda Bear who is 3 years old
Ming Ming is a Female Panda Bear who is 8 years old
============================
Pan Pan sleeps 10 hours daily and should have Broccoli Chicken
Kunfu sleeps 4 hours daily and should have Mixed Veggies
Ming Ming's duration is unknown thus shouldhave only bamboo leaves
```

In [8]:

```
#TASK - 04

class Cat:
    def __init__(self, color="White", action="sitting") -> None:
        self.color, self.action = color, action

    def printCat(self):
        print(self.color, "cat is", self.action)
    def changeColor(self, color):
        self.color = color

c1 = Cat()
c2 = Cat("Black")
c3 = Cat("Brown", "jumping")
c4 = Cat("Red", "purring")
c1.printCat()
c2.printCat()
c3.printCat()
c4.printCat()
c1.changeColor("Blue")
c3.changeColor("Purple")
c1.printCat()
c3.printCat()
```

```
White cat is sitting
Black cat is sitting
Brown cat is jumping
Red cat is purring
Blue cat is sitting
Purple cat is jumping
```

In [9]:

```python
#TASK - 05

class Student:
    def __init__(self, name="dafault student") -> None:
        self.name=name
    def quizcalc(self, q1=0,q2=0,q3=0):
        self.score =  (q1+q2+q3)/3
    def printdetail(self):
        print(f"Hello {self.name}\nYour average quiz score is {self.score}")

s1 = Student()
s1.quizcalc(10)
print('---------------------------------')
s1.printdetail()
s2 = Student('Harry')
s2.quizcalc(10,8)
print('---------------------------------')
s2.printdetail()
s3 = Student('Hermione')
s3.quizcalc(10,9,10)
print('---------------------------------')
s3.printdetail()
```

```
---------------------------------
Hello dafault student
Your average quiz score is 3.3333333333333335
---------------------------------
Hello Harry
Your average quiz score is 6.0
---------------------------------
Hello Hermione
Your average quiz score is 9.666666666666666
```

In [10]:

```python
#TASK - 06

class Vehicle:
    def __init__(self) -> None:
        self.position = [0,0]
    def moveUp(self): self.position[1]+=1
    def moveDown(self): self.position[1]-=1
    def moveRight(self): self.position[0]+=1
    def moveLeft(self): self.position[0]-=1

    def print_position(self):
        print(tuple(self.position))

car = Vehicle()
car.print_position()
car.moveUp()
car.print_position()
car.moveLeft()
car.print_position()
car.moveDown()
car.print_position()
car.moveRight()
```

```
(0, 0)
(0, 1)
(-1, 1)
(-1, 0)
```

In [11]:

```python
#TASK - 07

class Programmer:
    def __init__(self, name, lang, exp) -> None:
        self.name, self.lang, self.exp = name, lang, exp
        print("Horray! A new programmer is born")
    def addExp(self, exp):
        print(f"Updating experience of {self.name}")
        self.exp += exp
    def printDetails(self):
        print(f"Name: {self.name}\nLanguage: {self.lang}\nExperience: {self.

p1 = Programmer("Ethen Hunt", "Java", 10)
p1.printDetails()
print('--------------------------')
p2 = Programmer("James Bond", "C++", 7)
p2.printDetails()
print('--------------------------')
p3 = Programmer("Jon Snow", "Python", 4)
p3.printDetails()
p3.addExp(5)
p3.printDetails()
```

```
Horray! A new programmer is born
Name: Ethen Hunt
Language: Java
Experience: 10 years.
--------------------------
Horray! A new programmer is born
Name: James Bond
Language: C++
Experience: 7 years.
--------------------------
Horray! A new programmer is born
Name: Jon Snow
Language: Python
Experience: 4 years.
Updating experience of Jon Snow
Name: Jon Snow
Language: Python
Experience: 9 years.
```

In [13]:

```python
#TASK - 08

class Student:
    def __init__(self, name, Id, dept="CSE") -> None:
        self.name, self.id, self.dept, self.eff = name, Id, dept, None
    def dailyEffort(self, eff):
        self.eff = eff
        if eff <= 2: self.sugg = "Should give more effort!"
        elif eff <=4: self.sugg = "Keep up the good work!"
        else: self.sugg = "Excellent! Now motivate others"
    def printDetails(self):
        print(f"Name: {self.name}\nID: {self.id}\nDepartment: {self.dept}\nD
harry = Student('Harry Potter', 123)
harry.dailyEffort(3)
harry.printDetails()
print('========================')
john = Student("John Wick", 456, "BBA")
john.dailyEffort(2)
john.printDetails()
print('========================')
naruto = Student("Naruto Uzumaki", 777, "Ninja")
naruto.dailyEffort(6)
naruto.printDetails()
```

```
Name: Harry Potter
ID: 123
Department: CSE
Daily Effort: 3 hour(s)
Suggestion: Keep up the good work!
========================
Name: John Wick
ID: 456
Department: BBA
Daily Effort: 2 hour(s)
Suggestion: Should give more effort!
========================
Name: Naruto Uzumaki
ID: 777
Department: Ninja
Daily Effort: 6 hour(s)
Suggestion: Excellent! Now motivate others
```

In [16]:

```
#TASK - 09

class Patient:
    def __init__(self, name, age):
        self.name, self.age = name, age
    def add_Symptom(self, *sysmptoms):
        self.sysmptoms = ", ".join(sysmptoms)
    def printPatientDetail(self):
        print(f"Name: {self.name}\nAge: {self.age}\nSymptoms: {self.sysmptom


p1 = Patient("Thomas", 23)
p1.add_Symptom("Headache")
p2 = Patient("Carol", 20)
p2.add_Symptom("Vomiting", "Coughing")
p3 = Patient("Mike", 25)
p3.add_Symptom("Fever", "Headache", "Coughing")
print("=========================")
p1.printPatientDetail()
print("=========================")
p2.printPatientDetail()
print("=========================")
p3.printPatientDetail()
print("=========================")
```

```
=========================
Name: Thomas
Age: 23
Symptoms: Headache
=========================
Name: Carol
Age: 20
Symptoms: Vomiting, Coughing
=========================
Name: Mike
Age: 25
Symptoms: Fever, Headache, Coughing
=========================
```

In [17]:

```
#TASK - 10

class Avengers:
    def __init__(self, name, partner): self.name, self.partner = name, partn
    def super_powers(self, *powers): self.powers = ", ".join(powers)
    def printAvengersDetail(self): print(f"Name: {self.name}\nPartner: {self

a1 = Avengers('Captain America', 'Bucky Barnes')
a1.super_powers('Stamina', 'Slowed ageing')
a2 = Avengers('Doctor Strange', 'Ancient One')
a2.super_powers('Mastery of magic')
a3 = Avengers('Iron Man', 'War Machine')
a3.super_powers('Genius level intellect', 'Scientist ')
print("=========================")
a1.printAvengersDetail()
print("=========================")
a2.printAvengersDetail()
print("=========================")
a3.printAvengersDetail()
print("=========================")
```

```
=========================
Name: Captain America
Partner: Bucky Barnes
Super powers: Stamina, Slowed ageing
=========================
Name: Doctor Strange
Partner: Ancient One
Super powers: Mastery of magic
=========================
Name: Iron Man
Partner: War Machine
Super powers: Genius level intellect, Scientist
=========================
```

In [18]:

```
#TASK - 11

class Shinobi:
    def __init__(self, name, rank):
        self.name, self.rank, self.mission, self.salary= name, rank.strip().
    def calSalary(self, n_missions):
        self.mission = n_missions
        if self.rank.title() == 'Genin': self.salary = self.mission * 50
        elif self.rank.title() == 'Chunin': self.salary = self.mission * 100
        else: self.salary = self.mission * 500
    def changeRank(self, rank): self.rank = rank.strip().title()
    def printInfo(self):
        print(f"Name: {self.name}\nRank: {self.rank}\nNumber of mission: {se


naruto = Shinobi("Naruto", "Genin")
naruto.calSalary(5)
naruto.printInfo()
print('====================')
shikamaru = Shinobi('Shikamaru', "Genin")
shikamaru.printInfo()
shikamaru.changeRank("Chunin")
shikamaru.calSalary(10)
shikamaru.printInfo()
print('====================')
neiji = Shinobi("Neiji", "Jonin")
neiji.calSalary(5)
neiji.printInfo()
```

```
Name: Naruto
Rank: Genin
Number of mission: 5
Salary: 250
====================
Name: Shikamaru
Rank: Genin
Number of mission: 0
Salary: 0
Name: Shikamaru
Rank: Chunin
Number of mission: 10
Salary: 1000
```

```
====================
Name: Neiji
Rank: Jonin
Number of mission: 5
Salary: 2500
```

In [19]:

```
#TASK - 12

class ParcelKoro:
    def __init__(self, name="No name set", product_weight=0):
        self.name = name
        self.product_weight = product_weight
        self.product_fee = 0
    def calculateFee(self, location=None):
        if self.product_weight:
            if location: self.product_fee = (self.product_weight*20)+100
            else: self.product_fee = (self.product_weight*20)+50

    def printDetails(self):
        print(f"Customer Name: {self.name}\nProduct Weight: {self.product_we


print("**********************")
p1 = ParcelKoro()
p1.calculateFee()
p1.printDetails()
print("**********************")
p2 = ParcelKoro('Bob The Builder')
p2.calculateFee()
p2.printDetails()
print("----------------------------")
p2.product_weight = 15
p2.calculateFee()
p2.printDetails()
print("**********************")
p3 = ParcelKoro('Dora The Explorer', 10)
p3.calculateFee('Dhanmondi')
p3.printDetails()
```

```
**********
Customer Name: No name set
Product Weight: 0
Total fee: 0
**********
Customer Name: Bob The Builder
Product Weight: 0
```

```
Total fee: 0
----------------------------
Customer Name: Bob The Builder
Product Weight: 15
Total fee: 350
**********************
Customer Name: Dora The Explorer
Product Weight: 10
Total fee: 300
```

In [2]:

```
#TASK - 13

class Batsman:
    def __init__(self, *var):
        if len(var) == 2:
            runs, balls = var
        else:
            player, runs, balls = var

        if len(var) != 2:
            self.name = player
        else:
            self.name = "New Batsman"

        self.runs = runs
        self.balls = balls

    def setName(self, name):
        self.name = name

    def battingStrikeRate(self):
        return (self.runs) / (self.balls) * 100

    def printCareerStatistics(self):
        print(f"Name: {self.name}")
        print(f"Runs Scored: {self.runs}, Balls Faced: {self.balls}")



b1 = Batsman(6101, 7380)
b1.printCareerStatistics()
print("============================")
b2 = Batsman("Liton Das", 678, 773)
b2.printCareerStatistics()
print("----------------------------")
print(b2.battingStrikeRate())
print("============================")
b1.setName("Shakib Al Hasan")
```

```
b1.printCareerStatistics()
print("----------------------------")
print(b1.battingStrikeRate())
```

```
Name: New Batsman
Runs Scored: 6101, Balls Faced: 7380
============================
Name: Liton Das
Runs Scored: 678, Balls Faced: 773
----------------------------
87.71021992238033
============================
Name: Shakib Al Hasan
Runs Scored: 6101, Balls Faced: 7380
----------------------------
82.66937669376694
```

In [23]:

```
#TASK - 14

class EPL_Team:
    def __init__(self, team, song="No Slogan") -> None:
        self.team, self.song, self.n_title = team, song, 0
    def increaseTitle(self, n=1):
        self.n_title += n
    def changeSong(self, song):
        self.song = song
    def showClubInfo(self):
        return f"Name: {self.team}\nSong: {self.song}\nTotal No of title: {s


manu = EPL_Team('Manchester United', 'Glory Glory Man United')
chelsea = EPL_Team('Chelsea')
print('====================')
print(manu.showClubInfo())
print('####################')
manu.increaseTitle()
print(manu.showClubInfo())
print('====================')
print(chelsea.showClubInfo())
chelsea.changeSong('Keep the blue flag flying high')
print(chelsea.showClubInfo())
```

```
====================
Name: Manchester United
Song: Glory Glory Man United
Total No of title: 0
####################
Name: Manchester United
Song: Glory Glory Man United
Total No of title: 1
====================
Name: Chelsea
Song: No Slogan
Total No of title: 0
Name: Chelsea
Song: Keep the blue flag flying high
Total No of title: 0
```

In [24]:

```
#TASK - 15

class Account:
    def __init__(self, name="Default Account", balance=0.0) -> None:
        self.name, self.balance = name, balance

    def withdraw(self, amount):
        if self.balance-amount <= 3070:
            print("Sorry, Withdraw unsuccessful! The account balance after d
        else:
            self.balance-=amount
            print(f"Withdraw successful! New balance is: {self.balance}")
    def details(self):
        return f"{self.name}\n{self.balance}"


a1 = Account()
print(a1.details())
print("------------------------")
a1.name = "Oliver"
a1.balance = 10000.0
print(a1.details())
print("------------------------")
a2 = Account("Liam")
print(a2.details())
print("------------------------")
a3 = Account("Noah",400)
print(a3.details())
print("------------------------")
a1.withdraw(6930);
print("------------------------")
a2.withdraw(600);
print("------------------------")
a1.withdraw(6929)
```

```
Default Account
0.0
------------------------
Oliver
```

```
10000.0
------------------------
Liam
0.0
------------------------
Noah
400
------------------------
Sorry, Withdraw unsuccessful! The account balance after deducting withdraw amount is equal to or less t
------------------------
Sorry, Withdraw unsuccessful! The account balance after deducting withdraw amount is equal to or less t
------------------------
Withdraw successful! New balance is: 3071.0
```

In [2]:

```python
#TASK - 16

class Author:
    def __init__(self, name="Default", *books) -> None:
        self.name, self.books = name, books
    def addBooks(self, *books): self.books += books
    def changeName(self, name): self.name = name

    def printDetails(self):
        print(f"Author Name: {self.name}\n--------\nList of Books:")
        for book in self.books: print(book)

auth1 = Author('Humayun Ahmed')
auth1.addBooks('Deyal', 'Megher Opor Bari')
auth1.printDetails()
print('====================')
auth2 = Author()
print(auth2.name)
auth2.changeName('Mario Puzo')
auth2.addBooks('The Godfather', 'Omerta', 'The Sicilian')
print('====================')
auth2.printDetails()
print('====================')
auth3 = Author('Paolo Coelho', 'The Alchemist', 'The Fifth Mountain')
auth3.printDetails()
```

```
Author Name: Humayun Ahmed
--------
List of Books:
Deyal
Megher Opor Bari
====================
Default
====================
Author Name: Mario Puzo
--------
List of Books:
The Godfather
Omerta
The Sicilian
====================
Author Name: Paolo Coelho
--------
List of Books:
```

```
The Alchemist
The Fifth Mountain
```