In [5]:

```
# 1.WAP to count the number of objects created.
class Object:
    count = 0
    def __init__(self):
        Object.count += 1

o1 = Object()
o2 = Object()
o3 = Object()
o4 = Object()
print("The total number of Objects are ",Object.count)
```

```
The total number of Objects are  4
```

In [7]:

```
# 2.WAP that has a class Student storing student information including DOB. The program should subtract Do
class Voter:
    def __init__(self,name,roll,DOB):
        self.name = name
        self.roll = roll
        self.DOB = DOB
        print("Name :-",name)
        print("Roll :-",roll)
        print("Date of birth :- ",DOB)
        if(DOB>2005):
            print("Not Eligible to vote")
        else:
            print("Eligible to vote")

obj1 = Voter("Bibek",20051722,2003)
```

```
Name :- Bibek
Roll :- 20051722
Date of birth :-  2003
Eligible to vote
```

In [15]:

```
# 3.WAP to enter a number, find the factorial of a number, check, whether the number is prime or not, find
import math
class Fun:
    def __init__(self,num):
        self.num = num
    def fact(self):
        if(self.num==1):
            return 1
        fac = 1
        for i in range (1,self.num+1):
            fac*=i
        return fac
    def prime(self):
        a = self.num
        for i in range (2,int(math.sqrt(a))+1):
            if(a%i == 0):
                print("It is a prime number")
                return 0
        print("It is not a prime")
    def sq(self):
        a = self.num
        return a**2
obj = Fun(7)
print("The factorial of the number is ",obj.fact())
obj.prime()
print("The square of the number is",obj.sq())
```

```
The factorial of the number is  5040
It is not a prime
The square of the number is 49
```

In [19]:

```
# 4.WAP that has a class circle. Use class variable to define the values of constant PI.Use class variable
class Circle:
    pi = 3.14
    def __init__(self,rad):
        self.r = rad
    def area(self):
        return self.pi * (self.r**2)
    def circum(self):
        return 2*self.pi*self.r
rad = float(input("Enter the value of radius:-"))
obj = Circle(rad)
print("The area of circle is ",obj.area())
print("The circumference of circle is",obj.circum())
```

```
Enter the value of radius:-28
The area of circle is  2461.76
The circumference of circle is 175.84
```

In [24]:

```python
# 5. WAP that has a class Student that stores a roll number, name, and marks in three subjects.Display in
class Student:
    def __init__(self,name,roll,mark1,mark2,mark3):
        self.name=name
        self.roll=roll
        self.mark1=mark1
        self.mark2=mark2
        self.mark3=mark3
    def printDetails(self):
        print("Name:",self.name)
        print("Roll:",self.roll)
        print("Total Score:",self.mark1+self.mark2+self.mark3)
n=input("Enter name:")
r=input("Enter roll:")
print("Enter marks in 3 subjects:")
m1=int(input())
m2=int(input())
m3=int(input())
s1=Student(n,r,m1,m2,m3)
s1.printDetails()
```

```
Enter name:Bibek
Enter roll:20051722
Enter marks in 3 subjects:
87
78
82
Name: Bibek
Roll: 20051722
Total Score: 247
```

In [25]:

```python
# 6. Write a class rectangle that has attribute length, breadth and a method area which return area of th
class Rect:
    def __init__(self,length,bre):
        self.l = length
        self.b = bre
    def area(self):
        return self.l * self.b
l = float(input("Enter the value of length:-"))
b = float(input("Enter the value of breadth:-"))
obj = Circle(l,b)
print("The area of rectangle is ",obj.area())
```

```
Enter the value of length:-12
Enter the value of breadth:-23
The area of rectangle is  276.0
```

In [5]:

```python
# 7. WAP to enter sides of different geometric figure, find the area of an 5 geometric figure.
class GeometryArea:
    def triangle(self,b,h):
        self.b = b
        self.h = h
        return 0.5*self.b*self.h

    def square(self,side):
        self.side = side
        return self.side*self.side

    def rectangle(self,length,breadth):
        self.length = length
        self.breadth = breadth
        return self.length*self.breadth

    def paralelogram(self,base,height):
        self.base = base
        self.height = height
        return self.base*self.height

    def circle(self,radius):
        self.radius = radius
        return 3.14*self.radius*self.radius

obj1= GeometryArea()
print("Area of triangle : ",obj1.triangle(9,4))
print("Area of rectangle ",obj1.rectangle(7,6))
print("Area of square ",obj1.square(7))
print("Area of paralelogram ",obj1.paralelogram(6,8))
print("Area of circle ",obj1.circle(21))
```

```
Area of triangle :  18.0
Area of rectangle  42
Area of square  49
Area of paralelogram  48
Area of circle  1384.74
```

In [13]:

```
# 8. WAP to program to calculate SI and CI with appreciate input given to the program eg p, r, t input.
class INTEREST:
    p=int(input("Enter the principal: "))
    t=int(input("Enter the time: "))
    r=int(input("Enter the rate: "))
    def si(self):
        print("Simple Interest is ", (self.p*self.r*self.t)/100)
    def ci(self):
        print("Compound Interest is ",(self.p*(1+(self.r)/100)**t))

obj = INTEREST()
obj.si()
obj.ci()
```

```
Enter the principal: 234
Enter the time: 2
Enter the rate: 1
Simple Interest is  4.68
Compound Interest is  328.20301476367996
```

In [14]:

```
# 9. WAP to enter the recipes ordered by the customer, in a restaurant, prepare the bill in a proper form
print("--------------------------")
menu = {
    "pizza": 12.99,
    "pasta": 8.99,
    "salad": 6.99,
    "burger": 9.99,
    "fries": 3.99,
    "soda": 1.99
}
order = []
while True:
    item = input("Enter an item, or 'done' to finish: ")
    item=item.lower()
    if item == "done":
        break
    elif item not in menu:
        print("Sorry, we don't have that item.")
    else:
        order.append(item)
total = sum([menu[item] for item in order])
print("--------------------------")
print("ORDER SUMMARY")
print("--------------------------")
for item in order:
    print(item.capitalize(), "\t$", menu[item])
print("--------------------------")
print("Total:\t\t$", total)
print("--------------------------")
print("Thank you for dining with us!")
```

```
--------------------------
Enter an item, or 'done' to finish: pizza
Enter an item, or 'done' to finish: salad
Enter an item, or 'done' to finish: fries
Enter an item, or 'done' to finish: done
--------------------------
ORDER SUMMARY
--------------------------
Pizza   $ 12.99
Salad   $ 6.99
Fries   $ 3.99
--------------------------
Total:          $ 23.97
--------------------------
Thank you for dining with us!
```

In [17]:

```
# 10. Write a menu driven program that keeps records of books and journal available in a library.
class Library:
    def _init_(self):
        self.bookname= ""
        self.author=""
    def getdata(self):
        self.bookname = input("Enter Name of the Book: ")
        self.author = input("Enter Author of the Book: ")
    def display(self):
        print("Name of the Book: " ,self.bookname)
        print("Author of the Book:" ,self.author)
        print("\n")
book=[ ]
ch = 'y'
while(ch== 'y'):
    print("0.Break\n1. Add New Book \n2.Display Books")
    resp = int(input("Enter your choice :"))
    if resp==0:
        break
    elif(resp==1):
        L=Library( )
        L.getdata( )
        book.append(L)
    elif(resp==2):
        for x in book:
            x.display( )
    else:
        print("Invalid input….")
        ch = input("Do you want continue….")
```

```
0.Break
1. Add New Book
2.Display Books
Enter your choice :1
Enter Name of the Book: Hamlet
Enter Author of the Book: W.Shakespear
0.Break
1. Add New Book
2.Display Books
Enter your choice :1
Enter Name of the Book: Fear
Enter Author of the Book: Charlie Higson
0.Break
1. Add New Book
2.Display Books
Enter your choice :0
```

In [18]:

```
# 11. WAP that uses a class attributes to define some defaults titles in a college. Display the name along
class College:
    dean_title = "Dr. "
    professor_title = "Prof."
    department = "Computer Science"

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

    def display_info(self):
        print("Name: ", self.name)
#         print("Title: ", self.dean_title)
        print("Department: ", self.department)
        deanname=input("enter the name od dean: ")
        print(f"Dean of {self.department} = {self.dean_title}{deanname}")
        print("\nProfessors:")
        p=int(input("enter the no of professor: "))
        professors = [input("enter name of professor") for i in range(p)]
        for professor in professors:
            print(self.professor_title, professor, " - ", self.department)
n=input("Enter the college name: ")
college = College(n)
college.display_info()
```

```
Enter the college name: Kalinga Institute of Industrial technology
Name:  Kalinga Institute of Industrial technology
Department:  Computer Science
enter the name od dean: Bhawani Shankar Mishra
Dean of Computer Science = Dr. Bhawani Shankar Mishra

Professors:
enter the no of professor: 2
enter name of professorAbishekh Ray
enter name of professor Kumar devdutta
Prof. Abishekh Ray  -  Computer Science
Prof.  Kumar devdutta  -  Computer Science
```

In [ ]:

```python
# Write a menu driven program to read, add, subtract, multiply, divide and transpose two matrices
import numpy as np
def read_matrix():
    rows = int(input("Enter number of rows: "))
    cols = int(input("Enter number of columns: "))
    print("Enter the matrix elements, row by row:")
    elements = []
    for i in range(rows):
        row = []
        for j in range(cols):
            elem = float(input(f"Enter element [{i}][{j}]: "))
            row.append(elem)
        elements.append(row)
    return np.array(elements)

def display_matrix(matrix):
    print("Matrix:")
    print(matrix)

def add_matrices(matrix1, matrix2):
    return matrix1 + matrix2

def subtract_matrices(matrix1, matrix2):
    return matrix1 - matrix2

def multiply_matrices(matrix1, matrix2):
    return np.dot(matrix1, matrix2)

def divide_matrices(matrix1, matrix2):
    return np.linalg.solve(matrix2, matrix1.T).T

def transpose_matrix(matrix):
    return matrix.T
print("Matrix Operations Menu")
print("----------------------")
print("1. Read Matrix")
print("2. Display Matrix")
print("3. Add Matrices")
print("4. Subtract Matrices")
print("5. Multiply Matrices")
print("6. Divide Matrices")
print("7. Transpose Matrix")
print("0. Exit")
while True:
    choice = int(input("Enter your choice: "))
    if choice == 0:
        break
    elif choice == 1:
        matrix = read_matrix()
    elif choice == 2:
        display_matrix(matrix)
    elif choice == 3:
        matrix2 = read_matrix()
        result = add_matrices(matrix, matrix2)
        display_matrix(result)
    elif choice == 4:
        matrix2 = read_matrix()
        result = subtract_matrices(matrix, matrix2)
        display_matrix(result)
    elif choice == 5:
        matrix2 = read_matrix()
        result = multiply_matrices(matrix, matrix2)
        display_matrix(result)
    elif choice == 6:
        matrix2 = read_matrix()
        result = divide_matrices(matrix, matrix2)
        display_matrix(result)
    elif choice == 7:
        result = transpose_matrix(matrix)
        display_matrix(result)
    else:
        print("Invalid choice. Try again.")
```

```
Matrix Operations Menu
----------------------
1. Read Matrix
2. Display Matrix
3. Add Matrices
4. Subtract Matrices
5. Multiply Matrices
6. Divide Matrices
7. Transpose Matrix
0. Exit
Enter your choice: 2
Matrix:
[[4. 5. 3.]
 [5. 3. 6.]]
Enter your choice: 3
Enter number of rows: 2
Enter number of columns: 3
Enter the matrix elements, row by row:
Enter element [0][0]: 1
Enter element [0][1]: 24
Enter element [0][2]: 5
Enter element [1][0]: 3
Enter element [1][1]: 5
Enter element [1][2]: 4
Matrix:
[[ 5. 29.  8.]
 [ 8.  8. 10.]]
Enter your choice: 4
Enter number of rows: 2
Enter number of columns: 3
Enter the matrix elements, row by row:
Enter element [0][0]: 1
Enter element [0][1]: 2
Enter element [0][2]: 3
Enter element [1][0]: 1
Enter element [1][1]: 2
Enter element [1][2]: 3
Matrix:
[[3. 3. 0.]
 [4. 1. 3.]]
```