

**Python Lab File**

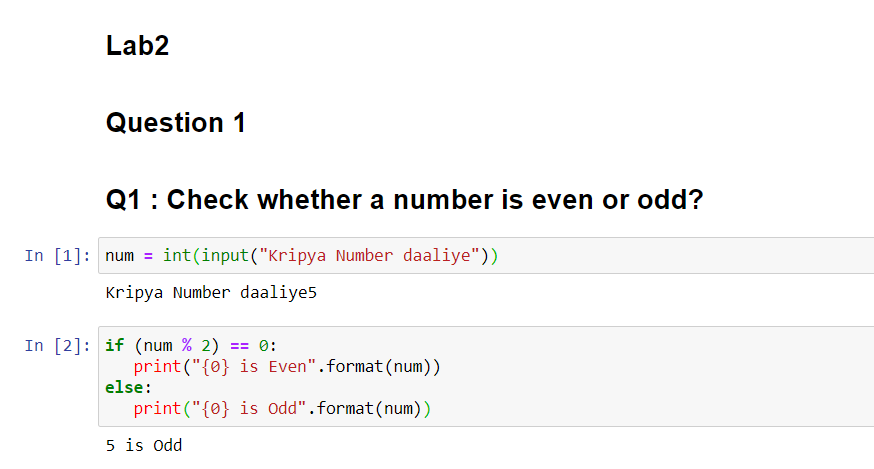
**Name: Parikshit Sharma**

**Class: Btech CSE 4C**

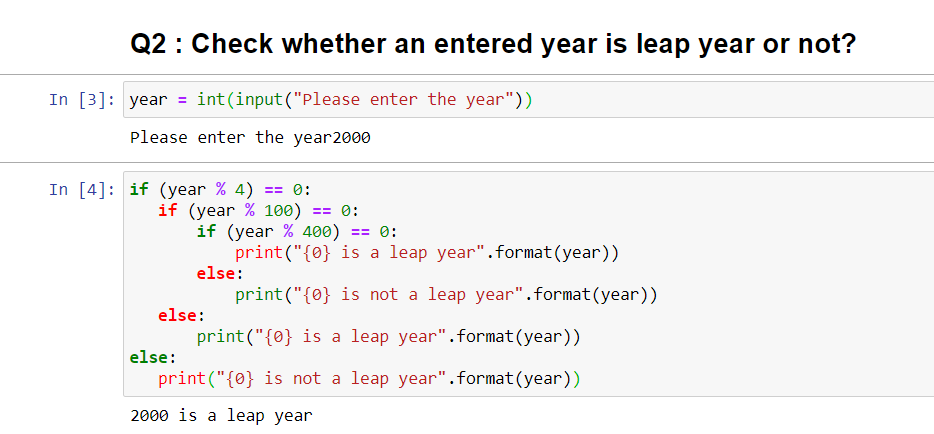
**Roll NO. 2K18CSUN01121**

**LAB 2**

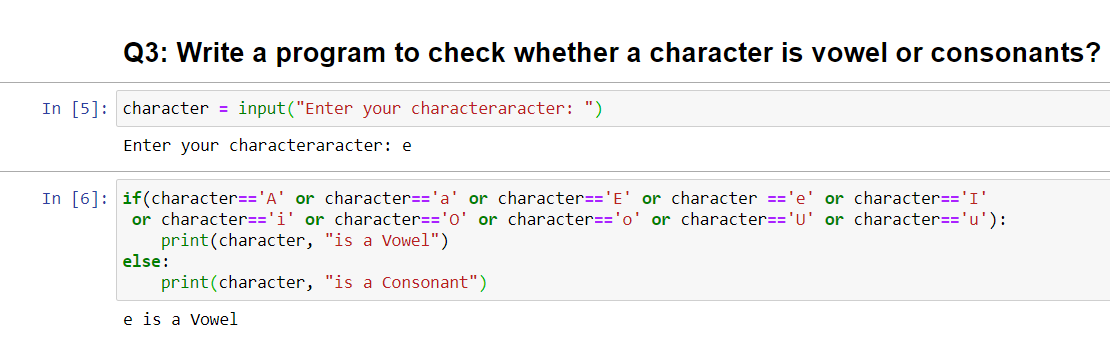
1. [Check whether a number is even or odd](https://www.programiz.com/python-programming/examples/prime-number).



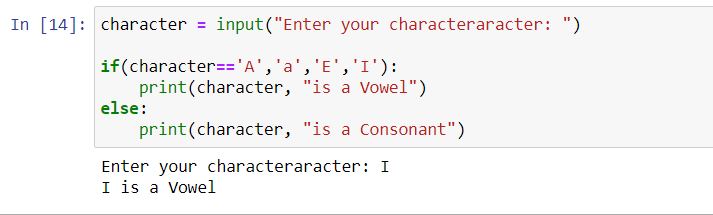
1. [Check whether an entered year is leap year or not.](https://www.programiz.com/python-programming/examples/prime-number)



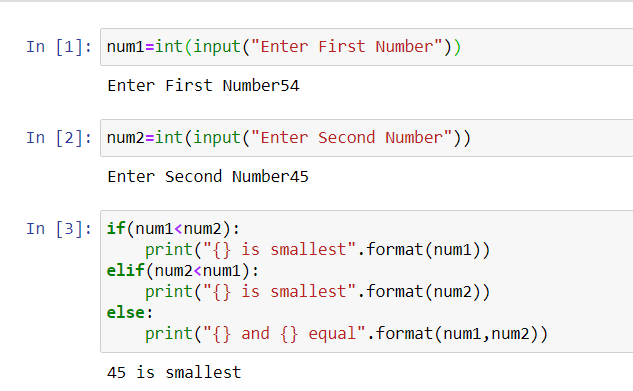
1. Write a program to check whether a character is vowel or consonants.



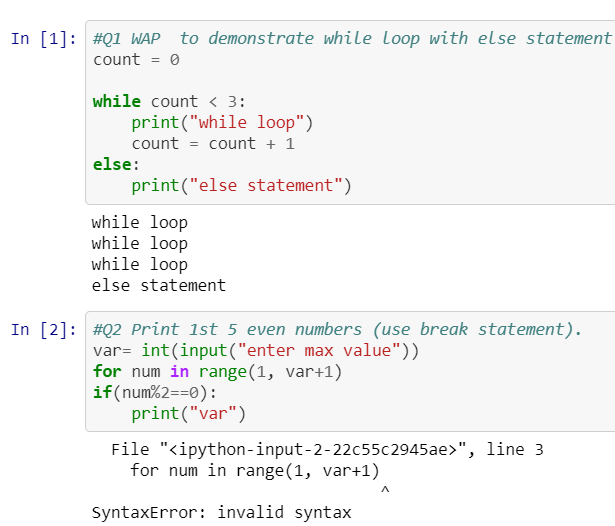
Alternative

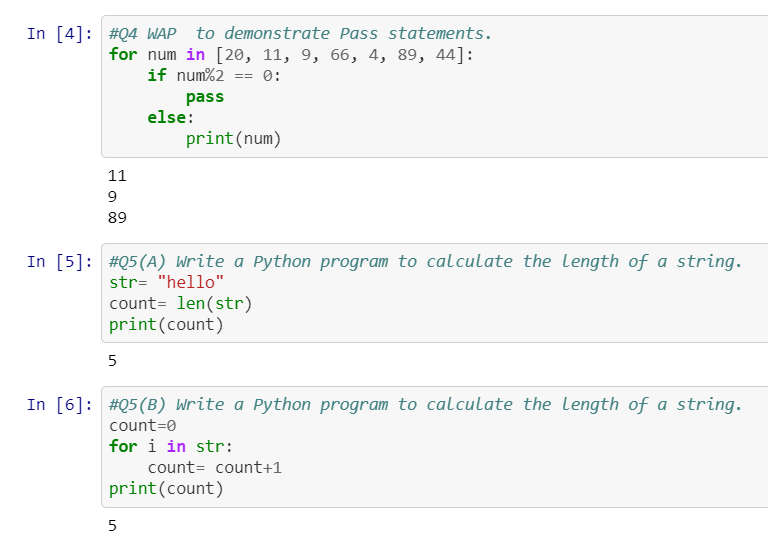


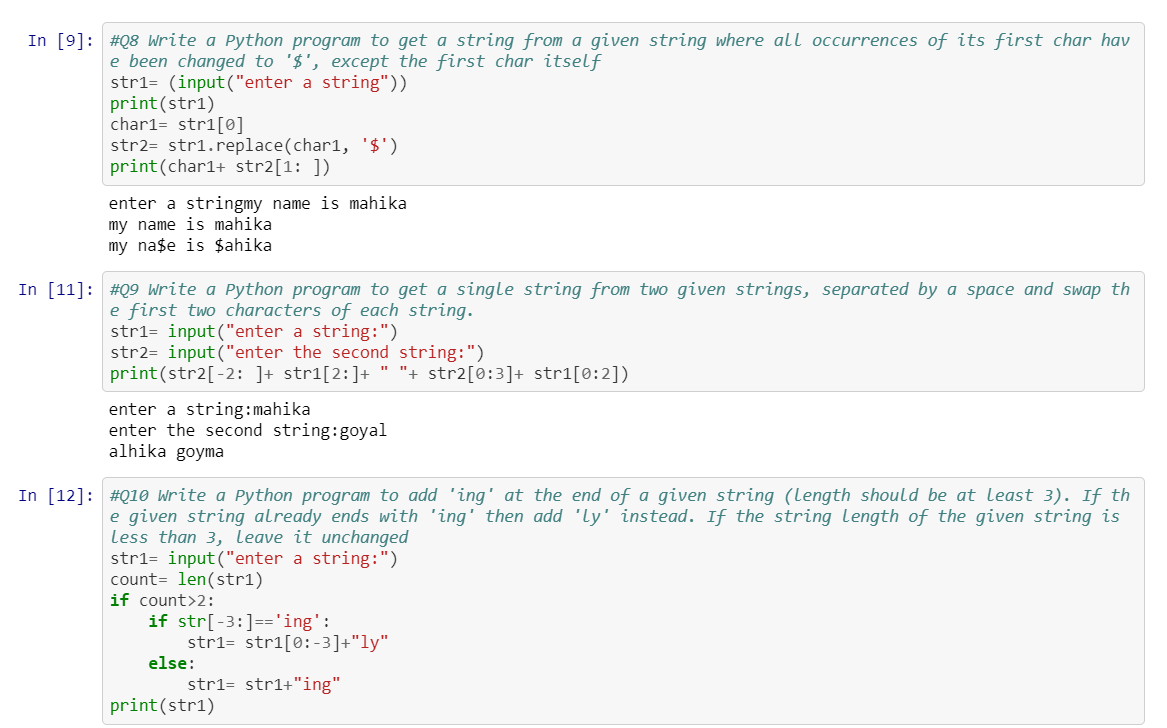
1. Write a program to find the smallest of two numbers?



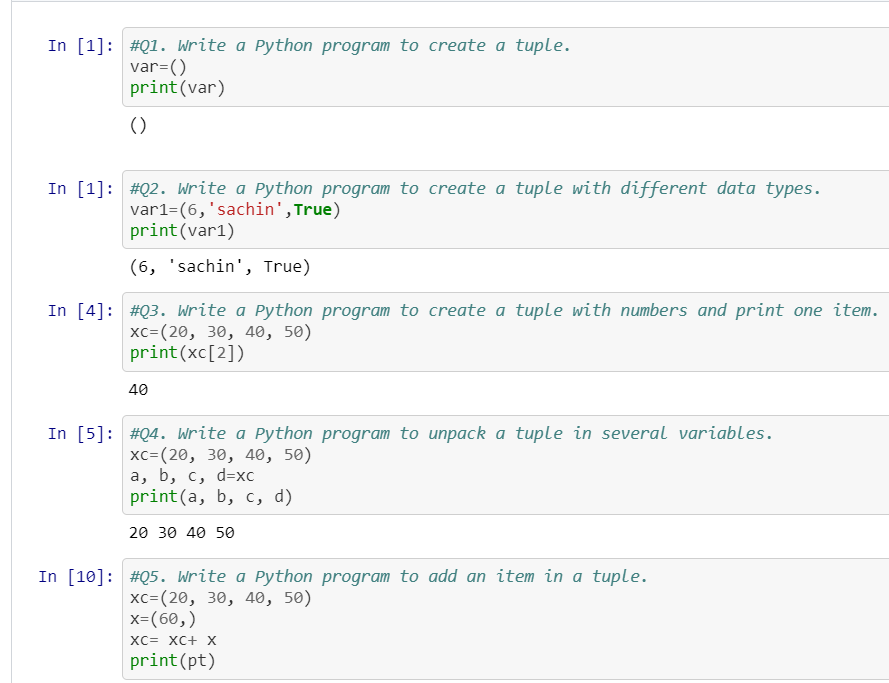
**LAB 3**

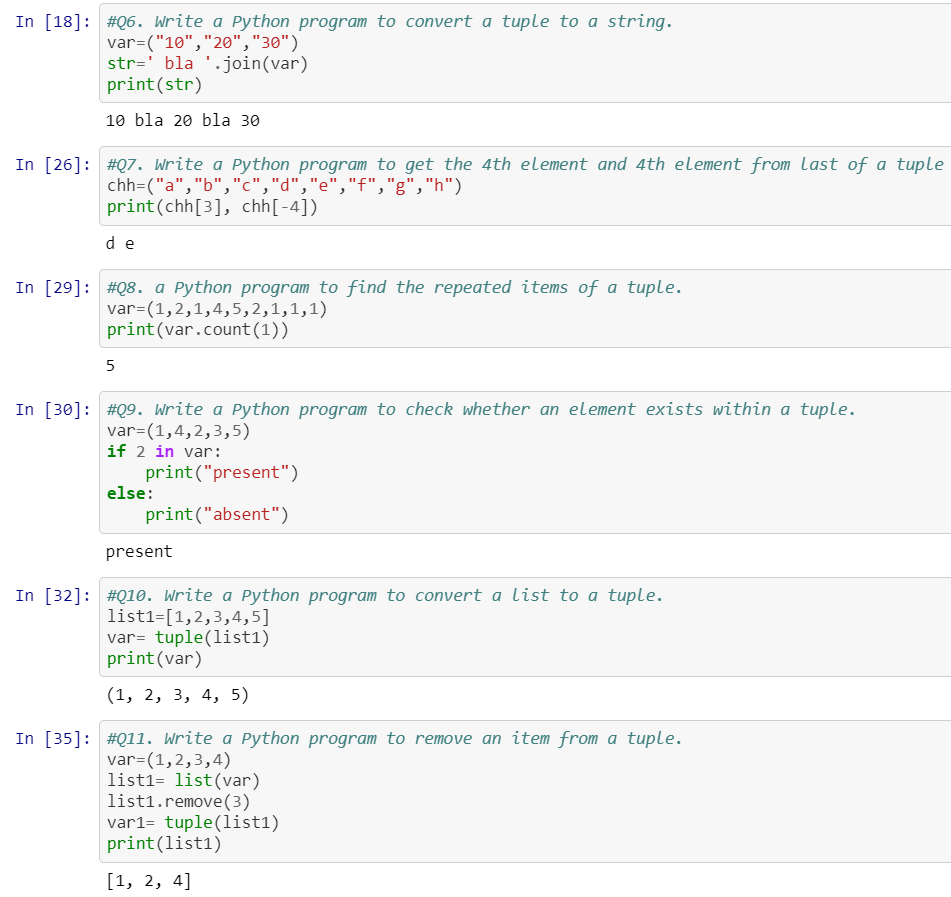






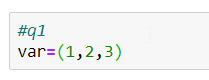
LAB 4

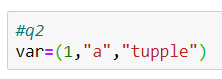


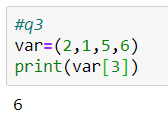


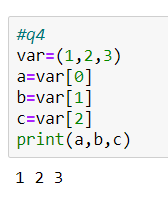


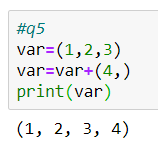
LAB 5,6

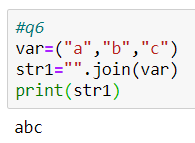


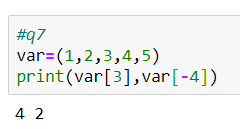


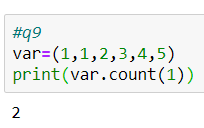


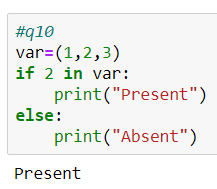


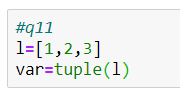


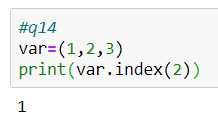


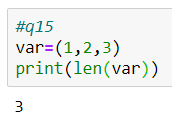












LAB 7

In [2]:

*#Q1*

**def** add\_num(a,b):*#function for multiplication*

multiply=a\*b;

**return** multiply; *#return value*

num1=int(input("input the number one: "))*#input from user for num1*

num2=int(input("input the number one: "))*#input from user for num2*

print("The product is",add\_num(num1,num2))*#call te function*

input the number one: 45

input the number one: 2

The product is 90

In [5]:

*#Q2*

**def** sum\_num(a,b):*#function for add*

add=a+b;

**return** add; *#return value*

num1=int(input("input the number one: "))*#input from user for num1*

num2=int(input("input the number one: "))*#input from user for num2*

print("The Sum is",sum\_num(num1,num2))*#call te function*

input the number one: 100

input the number one: 2

The product is 102

In [6]:

*#Q3*

**def** factorial(num):*#function definition*

fact=1

**for** i **in** range(1, num+1):*#for loop for finding factorial*

fact=fact\*i

**return** fact *#return factorial*

number=int(input("Please enter any number to find factorial: "))

result=factorial(number)*#function call and assign the value to variable result*

print("The factorial of **%d** = **%d**"%(number,result))

Please enter any number to find factorial: 5

The factorial of 5 = 120

In [9]:

*#Q4*

**def** recur\_fibo(n):

**if** n <= 1:

**return** n

**else**:

**return**(recur\_fibo(n-1) + recur\_fibo(n-2))

nterms = 10

*# check if the number of terms is valid*

**if** nterms <= 0:

print("Plese enter a positive integer")

**else**:

print("Fibonacci sequence:")

**for** i **in** range(nterms):

print(recur\_fibo(i))

Fibonacci sequence:

0

1

1

2

3

5

8

13

21

34

In [10]:

*#Q5*

**def** swap\_numbers(a, b):

temp = a

a = b

b = temp

print("After Swapping two Number: num1 = **{0}** and num2 = **{1}**".format(a, b))

num1 = float(input(" Please Enter the First Value : "))

num2 = float(input(" Please Enter the Second Value : "))

print("Before Swapping two Number: num1 = **{0}** and num2 = **{1}**".format(num1, num2))

swap\_numbers(num1, num2)

Please Enter the First Value : 4

Please Enter the Second Value : 2

Before Swapping two Number: num1 = 4.0 and num2 = 2.0

After Swapping two Number: num1 = 2.0 and num2 = 4.0

In [13]:

*#Q6*

**def** compute\_hcf(x, y):

**if** x > y:

smaller = y

**else**:

smaller = x

**for** i **in** range(1, smaller+1):

**if**((x % i == 0) **and** (y % i == 0)):

hcf = i

**return** hcf

num1 = int(input(" Please Enter the First Number : "))

num2 = int(input(" Please Enter the Second Number : "))

print("The H.C.F. is", compute\_hcf(num1, num2))

Please Enter the First Value : 34

Please Enter the Second Value : 54

The H.C.F. is 2

In [17]:

*#Q7*

c = (input(" Please Enter the Alphabet : "))

print("The ASCII value of '" + c + "' is", ord(c))

Please Enter the Alphabet : z

The ASCII value of 'z' is 122

In [21]:

*#Q8*

abs(-7)

Out[21]:

7

In [ ]:

any((1,0,0))

In [22]:

*#bin() converts an integer to a binary string*

bin(7)

Out[22]:

'0b111'

In [24]:

*#Q9*

**import** **datetime**

**from** **datetime** **import** date

print ("Present date is : ",end="")

print (date.today())

Present date is : 2020-03-30

In [27]:

*#Q10*

*# Function definition is here*

**def** printme( str ):

"This prints a passed string into this function"

print str

**return**;

*# Now you can call printme function*

printme("I'm first call to user defined function!")

printme("Again second call to the same function")

Hello , Good morning!

Hello , How do you do?

In [29]:

*#Q11*

*# Python program to demonstrate Keyword Arguments*

**def** student(firstname, lastname):

print(firstname, lastname)

*# Keyword arguments*

student(firstname ='Geeks', lastname ='Practice')

student(lastname ='Practice', firstname ='Geeks')

Geeks Practice

Geeks Practice

In [28]:

*#Q12*

*# Python program to demonstrate*

*# default arguments*

**def** myFun(x, y=50):

print("x: ", x)

print("y: ", y)

*# Driver code (We call myFun() with only*

*# argument)*

myFun(10)

x: 10

y: 50

In [ ]:

*#Q13*

*# Python program to illustrate*

*# \*args for variable number of arguments*

**def** myFun(\*argv):

**for** arg **in** argv:

print (arg)

myFun('Hello', 'Welcome', 'to', 'GeeksforGeeks')

LAB 8

|  |
| --- |
|  |
| #!/usr/bin/env python |
|  | # coding: utf-8 |
|  |  |
|  | # In[3]: |
|  |  |
|  |  |
|  | #1 |
|  |  |
|  | class triangle: |
|  | a=9 |
|  | b=4 |
|  | c=7 |
|  | st="Triangle created" |
|  |  |
|  | def create\_triangle(self): |
|  | print(self.st) |
|  | def print\_side(self): |
|  | print(self.a) |
|  | print(self.b) |
|  | print(self.c) |
|  |  |
|  | t=triangle() |
|  | t.print\_side() |
|  | t.create\_triangle() |
|  |  |
|  |  |
|  | # In[4]: |
|  |  |
|  |  |
|  | #2 |
|  | class IOString(): |
|  | def \_\_init\_\_(self): |
|  | self.str1 = "" |
|  |  |
|  | def get\_String(self): |
|  | self.str1 = input() |
|  |  |
|  | def print\_String(self): |
|  | print(self.str1.upper()) |
|  |  |
|  | str1 = IOString() |
|  | str1.get\_String() |
|  | str1.print\_String() |
|  |  |
|  |  |
|  | # In[5]: |
|  |  |
|  |  |
|  | #3 |
|  | class rectangle(): |
|  | def \_\_init\_\_(self,breadth,length): |
|  | self.breadth=breadth |
|  | self.length=length |
|  | def area(self): |
|  | return self.breadth\*self.length |
|  | a=int(input("Enter length of rectangle: ")) |
|  | b=int(input("Enter breadth of rectangle: ")) |
|  | obj=rectangle(a,b) |
|  | print("Area of rectangle:",obj.area()) |
|  |  |
|  | print() |
|  |  |
|  |  |
|  | # In[6]: |
|  |  |
|  |  |
|  |  |
|  | #4 |
|  | import math |
|  | class circle(): |
|  | def \_\_init\_\_(self,radius): |
|  | self.radius=radius |
|  | def area(self): |
|  | return math.pi\*(self.radius\*\*2) |
|  | def perimeter(self): |
|  | return 2\*math.pi\*self.radius |
|  |  |
|  | r=int(input("Enter radius of circle: ")) |
|  | obj=circle(r) |
|  | print("Area of circle:",round(obj.area(),2)) |
|  | print("Perimeter of circle:",round(obj.perimeter(),2)) |
|  |  |
|  |  |
|  | # In[7]: |
|  |  |
|  |  |
|  | #5 |
|  | class Class1: |
|  | def m(self): |
|  | print("In Class1") |
|  |  |
|  | class Class2(Class1): |
|  | def m(self): |
|  | print("In Class2") |
|  |  |
|  | class Class3(Class1): |
|  | def m(self): |
|  | print("In Class3") |
|  |  |
|  | class Class4(Class2, Class3): |
|  | def m(self): |
|  | print("In Class4") |
|  |  |
|  | obj = Class4() |
|  | obj.m() |
|  |  |
|  | Class2.m(obj) |
|  | Class3.m(obj) |
|  | Class1.m(obj) |
|  |  |
|  |  |
|  | # In[ ]: |
|  |  |

LAB 9

1.

class Triangle:

def \_\_init\_\_(self):

self.a = 0

self.b = 0

self.c = 0

def create\_triangle(self):

self.a = int(input("Enter the first side"))

self.b = int(input("Enter the second side"))

self.c = int(input("Enter the third side"))

def print\_sides(self):

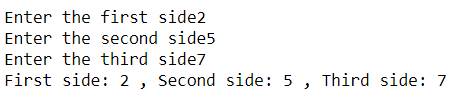
print("First side:",self.a,", Second side:",self.b,", Third side:",self.c)

x = Triangle()

x.create\_triangle()

x.print\_sides()

OutPut:



2.

class String():

def \_\_init\_\_(self):

self.str1 = ""

def inputstr(self):

self.str1 = input("Enter the string")

def printstr(self):

print(self.str1)

x = String()

x.inputstr()

x.printstr()

OutPut:



3.

class Rectangle:

def \_\_init\_\_(self,length,width):

self.length= length

self.width= width

def perimeter(self):

print("The perimeter is : ",2\*(self.length+self.width))

x = Rectangle(2,3)

x.perimeter()

13067

OutPut:

****

4.

class Circle:

def \_\_init\_\_(self,radius):

self.radius= radius

def area(self):

print("The area is: ", 3.14\*self.radius\*self.radius)

def perimeter(self):

print("The perimeter is: ", 2\*3.14\*self.radius)

x=Circle(8)

x.area()

x.perimeter()

OutPut:



5.

class A:

def m(self):

print("A")

class B:

def n(self):

print("B")

class C(A,B):

def o(self):

print("C")

x=C()

x.m()

x.n()

x.o()

OutPut:

****

LAB 10

|  |
| --- |
| # In[ ]: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | # In[9]: |
|  |  |
|  |  |
|  | ##Q3 |
|  | class MyClass: |
|  | "This is my second class" |
|  | a = 10 |
|  | def func(self): |
|  | print('Hello') |
|  |  |
|  | print(MyClass.a) |
|  | print(MyClass.func) |
|  | print(MyClass.\_\_doc\_\_) |
|  |  |
|  |  |
|  | # In[8]: |
|  |  |
|  |  |
|  | ##Q4 |
|  | class Parent(): |
|  |  |
|  | # Constructor |
|  | def \_\_init\_\_(self): |
|  | self.value = "Inside Parent" |
|  |  |
|  | # Parent show method |
|  | def show(self): |
|  | print(self.value) |
|  |  |
|  | # Define child class |
|  | class Child(Parent): |
|  |  |
|  | # Constructor |
|  | def \_\_init\_\_(self): |
|  | self.value = "Inside Child" |
|  |  |
|  | # Child's show method |
|  | def show(self): |
|  | print(self.value) |
|  |  |
|  |  |
|  | # Driver code |
|  | obj1 = Parent() |
|  | obj2 = Child() |
|  |  |
|  | obj1.show() |
|  | obj2.show() |
|  |  |
|  |  |
|  | # In[7]: |
|  |  |
|  |  |
|  | ##Q5 |
|  | class Class1: |
|  | def m(self): |
|  | print("In Class1") |
|  |  |
|  | class Class2(Class1): |
|  | def m(self): |
|  | print("In Class2") |
|  |  |
|  | class Class3(Class1): |
|  | def m(self): |
|  | print("In Class3") |
|  |  |
|  | class Class4(Class2, Class3): |
|  | def m(self): |
|  | print("In Class4") |
|  |  |
|  | obj = Class4() |
|  | obj.m() |
|  |  |
|  | Class2.m(obj) |
|  | Class3.m(obj) |
|  | Class1.m(obj) |
|  |  |
|  |  |
|  | # In[ ]: |