

I. COURSE OUTCOME 1(CO1)

PROGRAM NO: 1

DATE: 24/11/2021

AIM: Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE like PyCharm, PyDev...

Comparison between IDLE and Thonny

Thonny is built for education and you can download the latest version from the Thonny website. The download options are at the top right. Thonny looks quite different to IDLE - it has different panels for the editor, the shell and the variables watcher plus (show view) lots of other options as well. It has a powerful debugger built in and other tools which let you manage packages and plugins.

The Idle editor comes built-in with Python and is the one that many tutorials use by default. It's a fine, basic, editor that also has a Python shell built in for interactive programming. When you start Idle up, you get the shell window. This allows you to execute python commands and see the results immediately without having to create a program. This can be useful for trying things out.

PROGRAM NO: 2

DATE: 24/11/2021

AIM: Display future leap years from current year to a final year entered by user.

INPUT :

```
s=int(input("enter start year"))
```

```
e=int(input("enter end year:"))
```

```
if(s<e):
```

```
    print("leap years are:",end="")
```

```
for i in range(s,e):
```

```
    if i%4==0 and i%100!=0:
```

```
print(i,end=" ")
```

OUTPUT :

enter start year :2000

enter end year: 2500

leap years are: 2004 2008 2012 2016 2020 2024 2028 2032 2036 2040 2044 2048 2052 2056 2060
2064 2068 2072 2076 2080 2084 2088 2092 2096 2104 2108 2112 2116 2120 2124 2128 2132 2136
2140 2144 2148 2152 2156 2160 2164 2168 2172 2176 2180 2184 2188 2192 2196 2204 2208 2212
2216 2220 2224 2228 2232 2236 2240 2244 2248 2252 2256 2260 2264 2268 2272 2276 2280 2284
2288 2292 2296 2304 2308 2312 2316 2320 2324 2328 2332 2336 2340 2344 2348 2352 2356 2360
2364 2368 2372 2376 2380 2384 2388 2392 2396 2404 2408 2412 2416 2420 2424 2428 2432 2436
2440 2444 2448 2452 2456 2460 2464 2468 2472 2476 2480 2484 2488 2492 2496 2888 2892 2896
2904 2908 2912 2916 2920 2924 2928 2932 2936 2940 2944 2948 2952 2956 2960 2964 2968 2972
2976 2980 2984 2988 2992 2996

PROGRAM NO: 3

DATE:24/11/2021

AIM: List comprehensions:

A. Generate positive list of numbers from a given list of integers

INPUT :

```
list1 =[1,-1,2,-2,3,-3]
```

```
positive=[num for num in list1 if num>=0]
```

```
print(positive)
```

OUT PUT:

```
[1,2,3]
```

B. Square of N number

INPUT :

```
n=int(input("enter limit:"))
```

```
squarelist=[i**2 for i in range(1,n+1)]  
  
print("square of N numbers:",squarelist)
```

OUT PUT:

```
enter limit:5  
  
square of N numbers: [1, 4, 9, 16, 25]
```

C. Form a list of vowels selected from a given word**INPUT :**

```
word=str(input("enter the word:"))  
  
print("the original string is:"+word)  
  
print("the vowels are :",end="")  
  
for i in word:  
  
    if i in 'aeiouAEIOU':  
  
        print([i],end="")
```

OUT PUT:

```
enter the word:word  
  
the original string is:word  
  
the vowels are :['o']
```

D. List ordinal value of each element of a word**INPUT :**

```
w=input("enter a word:")  
  
print("ordinal values corresponding to each element is")  
  
for i in w:  
  
    print(i,end=":")
```

```
print(ord(i),end=" ")
```

OUTPUT:

enter a word:parvathi

ordinal values corresponding to each element is

p:112 a:97 r:114 v:118 a:97 t:116 h:104 i:105

PROGRAM NO: 4**DATE:24/11/2021****AIM:** Count the occurrences of each word in a line of text**INPUT :**

```
str1=input("enter a string")
```

```
wordlist=str1.split()
```

```
count=[]
```

```
for w in wordlist:
```

```
    count.append(wordlist.count(w))
```

```
print("count of occurrence:"+str(list(zip(wordlist,count))))
```

OUTPUT:

enter a string parvathi

count of occurrence:[('parvathi', 1)]

PROGRAM NO: 5**DATE:24/11/2021****AIM:** Prompt the user for a list of integers. For all values greater than 100, store 'over' instead**INPUT :**

```
n=[]
```

```
s=int(input("enter a limit"))
```

```
print("enter {s} values")

for i in range (0,s):

    n.append(int(input()))

print("\nthe list is after asssigning:\n")

for i in range(0,len(n)):

    if n[i]>=100:

        print("over")

    else:

        print(n[i])
```

OUTPUT:

enter a limit5

enter {s} values

12

123

14

13

100

the list is after asssigning:

12

over

14

13

Over

PROGRAM NO: 6

DATE:24/11/2021

AIM:Store a list of first names. Count the occurrences of 'a' within the list

INPUT :

```
list=["a","b","a"]
```

```
occ=list.count("a")
```

```
print("count of occurrences of a:",occ)
```

OUTPUT:

count of occurrences of a : 2

PROGRAM NO: 7**DATE:24/11/2021**

AIM: Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both

INPUT :

```
lst=[1,3,5,7,9,11,34]
```

```
lst1=[5,13,45,7,20,65,1]
```

```
s=int(0)
```

```
c=int(0)
```

```
if(len(lst)==len(lst1)):
```

```
    print("lists are of same length")
```

```
else:
```

```
    print("lists have diff length")
```

```
for i in range(0,len(lst) and len(lst1)):
```

```
    s=s+lst[i]
```

```
    c=c+lst1[i]
```

```
if(s==c):
```

```
    print("equal sum")
```

```
else:
```

```
print("not same sum")

print("elements that matched are:")

l=[]

for i in range(0,len(lst)):

    for j in range(0,len(lst1)):

        if lst[i]==lst1[j]:

            l.append(lst[i] and lst1[j])

        else:

            continue

print(l)
```

OUTPUT:

lists are of same length

not same sum

elements that matched are:

[1, 5, 7]

PROGRAM NO: 8**DATE:24/11/2021**

AIM:Get a string from an input string where all occurrences of first character replaced with '\$', except first character. [eg: onion -> oni\$n]

INPUT :

```
str1=input("enter a string:")

char=str1[0]

str1=str1.replace(char,'$')

str1=char+str1[1:]

print(str1)
```

OUTPUT:

enter a string:malayalam

malayala\$

PROGRAM NO: 9

DATE:24/11/2021

AIM:Create a string from given string where first and last characters exchanged. [eg: python - > nythop]

INPUT :

```
str=input("enter a string")
```

```
new_str=str[-1:]+str[1:-1]+str[:1]
```

```
print("new string:",new_str)
```

OUTPUT

enter a string parvathi

new string: iarvathp

PROGRAM NO: 10

DATE:24/11/2021

AIM:Accept the radius from user and find area of circle

INPUT :

```
pi=3.14
```

```
r=float(input("input the radius of circle:"))
```

```
result=3.14*r**2
```

```
print("the area of circle with radius is:",result)
```

OUTPUT:

input the radius of circle:3

the area of circle with radius is: 28.26

PROGRAM NO: 11**DATE:29/11/2021****AIM:** Find biggest of 3 numbers entered**INPUT :**

```
x=int(input("enter 1st number"))
```

```
y=int(input("enter 2nd number:"))
```

```
z=int(input("enter 3rd number"))
```

```
if(x>y) and (x>z):
```

```
    largest=x
```

```
elif(y>x) and (y>z):
```

```
    largest=y
```

```
else:
```

```
    largest=z
```

```
print("largest no is",largest)
```

OUTPUT:

```
enter 1st number12
```

```
enter 2nd number:1
```

```
enter 3rd number122
```

```
largest no is 122
```

PROGRAM NO: 12**DATE:29/11/2021****AIM:**Accept a file name from user and print extension of that**INPUT :**

```
file=input("enter filename:")
```

```
f=file.split(".")  
  
print("extension of file is:"+f[-1])
```

OUTPUT:

```
enter filename:hai.python  
  
extension of file is:python
```

PROGRAM NO: 13**DATE:29/11/2021**

AIM:Create a list of colors from comma-separated color names entered by user.Display first and last colors.

INPUT :

```
a=[]  
  
for i in range(3):  
  
    b=input("enter the color")  
  
    a.append(b)  
  
print(a)  
  
print(a[0])  
  
print(a[2])
```

OUTPUT:

```
enter the color red  
  
enter the color blue  
  
enter the color black  
  
['red', 'blue', 'black']  
  
red  
  
black
```

PROGRAM NO: 14**DATE:29/11/2021****AIM:**Accept an integer n and compute n+nn+nnn**INPUT :**

```
n=int(input("enter a number:"))
```

```
x=int("%s"%n)
```

```
y=int("%s%s"%(n,n))
```

```
z=int("%s%s%s"%(n,n,n))
```

```
print("n+nn+nnn:",x+y+z)
```

OUTPUT:

```
enter a number:2
```

```
n+nn+nnn: 246
```

PROGRAM NO: 15**DATE:29/11/2021****AIM:**Print out all colors from color-list1 not contained in color-list2**INPUT :**

```
color_list_1=set(["white","pink","red","blue"])
```

```
color_list_2=set(["red","green","pink"])
```

```
print(color_list_1.difference(color_list_2))
```

OUTPUT:

```
{'blue', 'white'}
```

PROGRAM NO: 16**DATE:29/11/2021**

AIM:Create a single string separated with space from two strings by swapping the character at position 1.

INPUT :

```
a="python"
```

```
b="java"
```

```
p1=a[0]
```

```
p2=b[0]
```

```
c=b[0]+a[1:len(a)]+" "+a[0]+b[1:len(b)]
```

```
print(a[1:len(a)])
```

```
print(c)
```

OUTPUT:

```
ython
```

```
jython pava
```

PROGRAM NO: 17

DATE:29/11/2021

AIM:Sort dictionary in ascending and descending order

INPUT :

```
import operator
```

```
d={ 1:2,3:4,4:3,2:1,0:0}
```

```
print('original dictionary:',d)
```

```
sorted_d=sorted(d.items(),key=operator.itemgetter(1))
```

```
print('dictionary in ascending order by value',sorted_d)
```

```
sorted_d=dict(sorted(d.items(),key=operator.itemgetter(1),reverse=True))
```

```
print('dictionary in descending order by value:',sorted_d)
```

OUTPUT:

original dictionary: {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

dictionary in ascending order by value [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]

dictionary in descending order by value: {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}

PROGRAM NO: 18**DATE:29/11/2021****AIM:**Merge two dictionaries**INPUT :**

```
d1={'a':100,'b':200}
```

```
d2={'x':300,'y':200}
```

```
print("dictionary1=",d1)
```

```
print("dictionary2=",d2)
```

```
d=d1.copy()
```

```
d.update(d2)
```

```
print("merged dictionary:",d)
```

OUT PUT:

```
dictionary1=: {'a': 100, 'b': 200}
```

```
dictionary2=: {'x': 300, 'y': 200}
```

```
merged dictionary: {'a': 100, 'b': 200, 'x': 300, 'y': 200}
```

PROGRAM NO: 19**DATE:29/11/2021****AIM:**Find gcd of 2 numbers.**INPUT :**

```
x=int(input("enter 1st no"))
```

```
y=int(input("enter 2nd no"))

i=1

while(i<=x and i<=y):

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

        gcd=i

    i=i+1

print("gcd:",gcd)
```

OUTPUT:

```
enter 1st no3
enter 2nd no56
gcd: 1
```

PROGRAM NO:20**DATE:29/11/2021****AIM:**From a list of integers, create a list removing even numbers.**INPUT :**

```
num=[7,8,120,25,44,20,27]

print("original list:",num)

num=[x for x in num if x%2!=0]

print("list after remooving even nos",num)
```

OUTPUT:

```
original list: [7, 8, 120, 25, 44, 20, 27]

list after remooving even nos [7, 25, 27]
```

II. COURSE OUTCOME 2(CO)

PROGRAM NO: 1

DATE:1/12/2021

AIM:Program to find the factorial of a number

INPUT :

```
n=int(input('Enter a number : '))
```

```
f=1
```

```
for i in range(1,n+1): f=f*i
```

```
print ('Factorial of',n, '=',f)
```

OUT PUT:

Enter a number: 5

Factorial of 5 is: 120

PROGRAM NO: 2

DATE:1/12/2021

AIM: Generate Fibonacci series of N terms

INPUT :

```
n=int(input("enter the limit"))
```

```
a=0
```

```
b=1
```

```
sum=0
```

```
count=1
```

```
print("fibonacci series",end="")
```

```
while(count<=n):
```

```
    print(sum,end=" ")
```

```
    count+=1
```

```
    a=b
```

```
b=sum
sum=a+b
```

OUT PUT:

Enter the limit:5

Fibonacci series :0 1 1 2 3

PROGRAM NO: 3**DATE:1/12/2021****AIM:** Find the sum of all items in a list**INPUT :**

```
list1 = [10, 15, 20, 25, 30]
```

```
total = sum(list1)
```

```
print("Sum of list : ",total)
```

OUT PUT:

sum of list : 100

PROGRAM NO: 4**DATE:1/12/2021****AIM:**Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.**INPUT :**

```
from math import sqrt as s
```

```
for i in range(1000,1700):
```

```
    if s(i)==int(s(i)) and i%2==0:
```

```
        print(i,end=" ")
```

OUT PUT:

1024 1156 1296 1444 1600

PROGRAM NO:5**DATE:1/12/2021****AIM:**Display the given pyramid with step number accepted from user.**INPUT :**

```
rows=int(input("enter a number of rows"))
```

```
for i in range(1,rows+1):
```

```
    for j in range (1,i+1):
```

```
        print(i*j,end="")
```

```
    print()
```

OUT PUT:

Enter the the number of rows : 3

1

2 4

3 6 9

PROGRAM NO:6**DATE:1/12/2021****AIM:**Count the number of characters (character frequency) in a string**INPUT :**

```
test_str=str(input("Enter the string : "))
```

```
freq = { }
```

```
for i in test_str:
```

```
    if i in freq:
```

```
        freq[i] += 1
```

```
    else:
```

```
freq[i] = 1
```

```
print ("Count of all characters : "+ str(freq))
```

OUTPUT :

Enter a string : parvathi

Count of all characters : {'p': 1, 'a': 2, 'r': 1, 'v': 1, 't': 1, 'h': 1, 'i': 1}

PROGRAM NO:7**DATE:8/12/2021****AIM:**Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'**INPUT :**

```
str=input("enter a string")
```

```
print("inputted string is ",str)
```

```
if(str.endswith("ing")):
```

```
    str=str+'ly'
```

```
else:
```

```
    str=str+'ing'
```

```
print("the formatted string is",str)
```

OUT PUT :

Enter the string:python

Inputted string is: python

The formatted string is : pythoning

PROGRAM NO:8**DATE:8/12/2021****AIM:**Accept a list of words and return length of longest word**INPUT :**

```
a=[]
```

```
n=int(input("enter the number of elements in list"))

for x in range(0,n):

    element=input("enter element"+str(x+1))

    a.append(element)

max1=len(a[0])

temp=a[0]

for i in a:

    if(len(i)>max1):

        max1=len(i)

        temp=i

print("longest word",temp)

print("length of longest word:",max1)
```

OUT PUT :

Enter the number of elements n list :3

Enter the element 1:hai

Enter the element 2:hello

Enter the element 3:hi

Longest word: hello

Length of longest word: 5

PROGRAM NO:9

DATE:8/12/2021

AIM:Construct following pattern using nested loop

*

* *

* * *

* * * *

* * *

* *

*

INPUT :

```
n=int(input("enter the limit"))
```

```
for i in range(n):
```

```
    for j in range(i):
```

```
        print('*',end="")
```

```
for i in range(n,0,-1):
```

```
    for j in range(i):
```

```
        print('*',end="")
```

```
    print("")
```

OUT PUT:

Enter the limit :5

*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*

PROGRAM NO:10

DATE:8/12/2021

AIM:Generate all factors of a number. def print_factors(x):

INPUT :

```
a=int(input("enter number \n"))
```

```
def fact(fact=1):
```

```
    for i in range(1,a+1):
```

```
        fact=fact*i
```

```
    i=i+1
```

```
    print("fac",fact)
```

```
fact()
```

OUTPUT:

Enter number :5

Fac :120

PROGRAM NO:11

DATE:8/12/2021

AIM:Write lambda functions to find area of square, rectangle and triangle.

INPUT :

```
import math
```

```
t_area = lambda b,h :1/2*b*h
```

```
r_area = lambda l,b :l*b
```

```
s_area = lambda a : a*a
```

```
print("Area of Triangle :", t_area(4,2))
```

```
print("Area of Rectangle:", r_area(3,2))
```

```
print("Area of Square :", s_area(2))
```

OUTPUT:

Area of trangle : 4.0

Area of rectangle :6

Area of square : 4

III. COURSE OUTCOME 3(CO3)

PROGRAM NO: 1

DATE:15/12/2021

AIM:Work with built-in packages

A . Module math

INPUT:

```
import math
print(math.pi)
import math as m
print(m.pi)
print(".....")
from math import pi,sqrt
print("vaue of pi is",math.pi)
print("value of sqr root is",sqrt(4))
print(".....")
print("value of cos",math.cos(90))
print("value of sin",math.sin(90))
print("value of tan",math.tan(0))
print(".....")
```

OUT PUT:

```
>>> %Run 'module math.py'
3.141592653589793
.....
3.141592653589793
.....
Value of pi is 3.141592653589793
Value of square root is 3.0
.....
Value of sin(90) is 0.8939966636005579
Value of cos(90) is -0.4480736161291701
-0.4480736161291701
Value of tan(90) is -1.995200412208242
.....
>>>
```

B. Module time

INPUT:

```
import time

print("current time in second",time.time())

print("current time",time.ctime())

print("current time after 30 sec:",time.ctime(time.time()+30))

t=time.localtime()

print("time:",t)

print("current month",t.tm_mon)

print("current year",t.tm_year)

print("current day",t.tm_mday)

print("current hour",t.tm_hour)

print("current miniut",t.tm_min)

print("current second",t.tm_sec)
```

OUTPUT:

```
>>> %Run 'module time.py'

Current time in second : 1640014835.8148754
Current time : Mon Dec 20 21:10:35 2021
Current time after 30 seconds : Mon Dec 20 21:11:05 2021
time: time.struct_time(tm_year=2021, tm_mon=12, tm_mday=20, tm_hour=21, tm_min=10,
tm_sec=35, tm_wday=0, tm_yday=354, tm_isdst=0)
current year: 2021
current month: 12
current day: 20
current week day: 0
current Hour: 21
current Minute: 10
current Second: 35

>>>
```

C. Module calendar

INPUT:

```
import calendar

mm=int(input("enter month:"))
```

```
yy=int(input("enter the year:"))
print(calendar.month(yy,mm))
print(calendar.calendar(2015))
```

OUTPUT:

```
>>> %Run 'module calendar.py'

Enter month: 12
Enter year :2021
    December 2021
Mo Tu We Th Fr Sa Su
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

                2015

    January          February          March
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
      1  2  3  4      1 2 3 4 5 6 7 8      1 2 3 4 5 6 7 8
 5  6  7  8  9 10 11  9 10 11 12 13 14 15  9 10 11 12 13 14 15
12 13 14 15 16 17 18 16 17 18 19 20 21 22 16 17 18 19 20 21 22
26 27 28 29 30 31  23 24 25 26 27 28  23 24 25 26 27 28 29
                                     30 31

    April            May                June
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
      1  2  3  4  5      1 2 3      1 2 3 4 5 6 7
 6  7  8  9 10 11 12  4  5  6  7  8  9 10  8  9 10 11 12 13 14
13 14 15 16 17 18 19 11 12 13 14 15 16 17 15 16 17 18 19 20 21
20 21 22 23 24 25 26 18 19 20 21 22 23 24 22 23 24 25 26 27 28
27 28 29 30  25 26 27 28 29 30 31 29 30

    July            August              September
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
      1  2  3  4  5      1 2      1 2 3 4 5 6
 6  7  8  9 10 11 12  3  4  5  6  7  8  9  7  8  9 10 11 12 13
13 14 15 16 17 18 19 10 11 12 13 14 15 16 14 15 16 17 18 19 20
20 21 22 23 24 25 26 17 18 19 20 21 22 23 21 22 23 24 25 26 27
27 28 29 30 31  24 25 26 27 28 29 30  28 29 30

    October          November          December
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su
      1  2  3  4      1 2 3 4 5 6 7 8      1 2 3 4 5 6
 5  6  7  8  9 10 11  2  3  4  5  6  7  8  7  8  9 10 11 12 13
12 13 14 15 16 17 18  9 10 11 12 13 14 15 14 15 16 17 18 19 20
19 20 21 22 23 24 25 16 17 18 19 20 21 22 21 22 23 24 25 26 27
26 27 28 29 30 31  23 24 25 26 27 28 29 28 29 30 31

>>>
```


D. Module datetime**INPUT:**

```
import datetime

t=datetime.time(22,56,44)

print(t)

print("hour",t.hour)

print("min",t.minute)

print("se",t.second)

print("mis",t.microsecond)


print(".....")


d=datetime.date.today()

print(d)

print("year",d.year)

print("month",d.month)

print("day",d.day)


print(".....")


d1=datetime.date.today()

print(d1)

td=datetime.timedelta(days=2)

print(td)

d2=d1+td

print(d2)


print(".....")


dt=datetime.datetime.combine(d,t)


print(dt)
```

OUTPUT :

```
>>> %Run 'module datetime.py'
22:56:44
Hour : 22
Minute : 56
Second : 44
Microsecond : 0
.....

2021-12-20
Year 2021
Month 12
Day 20
.....

2021-12-20
2 days, 0:00:00
2021-12-22
.....

2021-12-20 22:56:44
>>>
```

E. Statistics module**INPUT :**

```
import statistics

print(statistics.mean([3,4,3]))

print(statistics.median([1, 3, 5, 7, 9, 11, 13]))

print(statistics.mode([1, 1, -3, 3, 7, -9]))

print(statistics.variance([1, 3, 5, 7, 9, 11]))

print(statistics.stdev([1, 3, 5, 7, 9, 11]))
```

OUTPUT:

3.3333333333333335

7

1

14

3.7416573867739413

F. Random module**INPUT :**

```
import random
```

```
random.seed(10)

print(random.random())

mylist = ["apple", "banana", "cherry"]

print(random.sample(mylist, k=2))

print(random.random())

mylist2 = ["apple", "banana", "cherry"]

random.shuffle(mylist2)

print(mylist2)

mylist3 = ["apple", "banana", "cherry"]

print(random.choice(mylist3))
```

OUTPUT:

0.5714025946899135

['banana', 'cherry']

0.5780913011344704

['cherry', 'banana', 'apple']

banana

PROGRAM NO: 2

DATE:15/12/2021

AIM:Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)

INPUT :

Package graphics

(1) circle.py

INPUT:

```
def perimeter(r):  
    print ("Perimeter : ",2*3.14*r)  
def area(r):  
    print ("Area : ",3.14*r*r)
```

(2) rectangle.py

INPUT:

```
def perimeter(l,b):  
    print ("Perimeter : ",2*(l+b))  
def area(l,b):  
    print ("Area : ",l*b)
```

Subpackage ThreeDgraphics

(1) cuboid.py

INPUT:

```
def perimeter(l,b,h):  
    print ("Perimeter : ",4*(l+b+h))  
def area(l,b,h):  
    print ("Area : ",2*l*b+2*l*h+2*h*b)
```

(2)sphere.py

INPUT:

```
def volume(r):  
    print ("Volume : ",(4/3)*3.14*r*r*r)  
def area(r):  
    print ("Surface Area : ",4*3.14*r*r)
```

(3)graphicsuse.py

INPUT:

```
from graphics import rectangle  
from graphics import circle
```

```
from graphics.ThreeDgraphics import cuboid
from graphics.ThreeDgraphics import sphere
l=int(input("Enter the length,l : "))
b=int(input("Enter the breadth,b : "))
rectangle.perimeter(l,b)
rectangle.area(l,b)

r=int(input("Enter the radius,r : "))
circle.perimeter(r)
circle.area(r)
l=int(input("Enter the length,l : "))
b=int(input("Enter the breadth,b : "))
h=int(input("Enter the height,h : "))
cuboid.perimeter(l,b,h)
cuboid.area(l,b,h)
r=int(input("Enter the radius,r : "))
sphere.volume(r)
sphere.area(r)
```

OUTPUT:

```
>>> %Run graphicsuse.py
Enter the length,l : 4
Enter the breadth,b : 5
Perimeter : 18
Area : 20
Enter the radius,r : 5
Perimeter : 31.400000000000002
Area : 78.5
Enter the length,l : 4
Enter the breadth,b : 5
Enter the height,h : 6
Perimeter : 60
Area : 148
Enter the radius,r : 5
Volume : 523.3333333333334
Surface Area : 314.0
>>> %Run graphicsuse.py
```

IV. COURSE OUTCOME 4(CO4)

PROGRAM NO: 1

DATE:9/1/2022

AIM: Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area

INPUT :

class rectangle:

```
def __init__(self,length,breadth):
```

```
    self.length=length
```

```
    self.breadth=breadth
```

```
def area(self):
```

```
    area=self.length*self.breadth
```

```
    print("Area=",area)
```

```
    return(area)
```

```
def perimeter(self):
```

```
    per=2*(self.length+self.breadth)
```

```
    print("Perimeter=",per)
```

```
print("First Rectangle:")
```

```
b1=rectangle(2,2)
```

```
a1=b1.area()
```

```
b1.perimeter()
```

```
print("\nSecond Rectangle:")
```

```
b2=rectangle(3,3)
```

```
a2=b2.area()
```

```
b2.perimeter()
```

```
if a1 > a2:
```

```
print("\nArea of first rectangle is larger")

else:

    print("\nArea of second rectangle is larger")
```

OUTPUT:

First Rectangle:

Area= 4

Perimeter= 8

Second Rectangle:

Area= 9

Perimeter= 12

Area of second rectangle is larger

PROGRAM NO: 2

DATE:9/1/2022

AIM: Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

INPUT:

```
class bank:
```

```
    bal=0
```

```
    def __init__(self,accno,name,ac_type,bal):
```

```
        self.accno=accno
```

```
        self.name=name
```

```
        self.ac_type=ac_type
```

```
        self.bal=bal
```

```
    def display(self):
```

```
        print("\nAccount Info:")
```

```
print("Account Number:",self.accno)
```

```
print("Account Name:",self.name)
```

```
print("Account Type:",self.ac_type)
```

```
print("Account Balance:",self.bal)
```

```
def deposit(self):
```

```
    dep=int(input("Enter the amount to deposit:"))
```

```
    self.bal=self.bal+dep
```

```
def withdraw(self):
```

```
    w=int(input("Enter the amount to withdraw:"))
```

```
    if w > self.bal:
```

```
        print("Insufficient Balance")
```

```
    else:
```

```
        self.bal=self.bal-w
```

```
        print("RS-",w,"Withdrawn successfully")
```

```
acc_no=int(input("Enter the Account Number:"))
```

```
acc_name=input("Enter the name:")
```

```
acc_type=input("Enter the account type-(savings/current):")
```

```
balance=int(input("Enter the initial balance:"))
```

```
b1=bank(acc_no,acc_name,acc_type,balance)
```

```
while(1):
```

```
    print("\n1.Account Info\n2.Deposit\n3.Withdraw\n4.Exit")
```

```
    opt=int(input("Select your option:"))
```

```
    if opt == 1:
```

```
        b1.display()
```

```
    elif opt == 2:
```

```
        b1.deposit()
```



```
elif opt == 3:
    b1.withdraw()
elif opt == 4:
    print("Exited")
    break
else:
    print("Invalid Option")
```

OUTPUT:

Enter the Account Number:1233

Enter the name: PARVTHI

Enter the account type-(savings/current):savings

Enter the initial balance:2000

1.Account Info

2.Deposit

.Withdraw

4.Exit

Select your option:2

Enter the amount to deposit:200

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account Info:

Account Number: 1233

Account Name: Alan

Account Type: savings

Account Balance: 2200

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:3

Enter the amount to withdraw:100

RS- 100 Withdrawn successfully

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account Info:

Account Number: 1233

Account Name: Alan

Account Type: savings

Account Balance: 2100

1.Account Info

2.Deposit

3.Withdraw

4.Exit

Select your option:4

Exited

PROGRAM NO: 3

DATE:9/1/2022

AIM: Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.

INPUT:

class rectangle:

```
    def __init__(self,l,b):
```

```
        self.__length=l
```

```
        self.__breadth=b
```

```
    def area(self):
```

```
        self.area=self.__length*self.__breadth
```

```
        print("Area=",self.area)
```

```
    def __lt__(self,second):
```

```
        if self.area < second.area:
```

```
            return True
```

```
        else:
```

```
            return False
```

```
print("first Rectangle:")
```

```
len1=int(input("Enter the length:"))
```

```
bread1=int(input("Enter the breadth:"))
```

```
obj1=rectangle(len1,bread1)
```

```
obj1.area()
```

```
print("\nSecond Rectangle:")
```

```
len2=int(input("Enter the length:"))
```

```
bread2=int(input("Enter the breadth:"))

obj2=rectangle(len2,bread2)

obj2.area()

if obj1 < obj2 :

    print("\nArea of second rectangle is larger:")

else:

    print("\nArea of first rectangle is larger:")
```

OUTPUT:

first Rectangle:

Enter the length:2

Enter the breadth:3

Area= 6

Second Rectangle:

Enter the length:2

Enter the breadth:2

Area= 4

Area of first rectangle is larger

PROGRAM NO: 4**DATE:9/1/2022**

AIM: Create a class Time with private attributes hour, minute and second. Overload '+' operator to

find sum of 2 time

INPUT:

class time:

```
def __init__(self,hour,minute,second):

    self.__hour=hour

    self.__minute=minute

    self.__second=second

def __add__(self,second):

    print("\nHour:",self.__hour + second.__hour)


    if self.__minute + second.__minute > 60:

        h1=(self.__minute + second.__minute)//60

        m1=(self.__minute + second.__minute)%60

        print("Minutes:",h1," hour ",m1," minutes")

    else:

        print("Minutes:",self.__minute + second.__minute)


    if self.__second+second.__second > 60:

        m1=(self.__second+second.__second)//60

        s1=(self.__second+second.__second)%60

        print("seconds:",m1," minutes ",s1," seconds")

    else:

        print("Seconds:",self.__second + second.__second)

hour1=int(input("Enter the hour:"))

minute1=int(input("Enter the minutes:"))

sec1=int(input("Enter the second:"))

obj1=time(hour1,minute1,sec1)

hour2=int(input("\nEnter the hour:"))

minute2=int(input("Enter the minutes:"))
```

```
sec2=int(input("Enter the second:"))  
  
obj2=time(hour2,minute2,sec2)  
  
obj1 + obj2
```

OUTPUT:

Enter the hour:3

Enter the minutes:44

Enter the second:35

Enter the hour:4

Enter the minutes:55

Enter the second:45

Hour: 7

Minutes: 1 hour 39 minutes

seconds: 1 minutes 20 seconds

PROGRAM NO:5**DATE:9/1/2022**

AIM: Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

INPUT:

class publisher:

```
    def __init__(self,pname):
```

```
        self.pname=pname
```

```
    def display(self):
```

```
        print("Publisher Name:",self.pname)

class book(publisher):

    def get(self,title,author):

        self.title=title

        self.author=author

    def display(self):

        print("Title Name:",self.title)

        print("Author Name:",self.author)

class python(book):

    def __init__(self,price,nop,pname):

        super().__init__(pname)

        self.price=price

        self.nop=nop

    def details(self):

        print("Price:",self.price)

        print("No of pages:",self.nop)

s1=python(450,72,"K D")

s1.get("Flames","K D")

s1.display()

s1.details()
```

OUTPUT:

Title Name: Flames

Author Name: K D

Price: 450

No of pages: 72

V. COURSE OUTCOME 5(CO5)

PROGRAM NO: 1

DATE:30/1/2022

AIM:Write a Python program to read a file line by line and store it into a list.

INPUT:

```
f1=open("sample.txt","w")
f1.write("This is my first line.\n This is my second line \n This is my third line")

f1=open("sample.txt","r")

ff=f1.readlines()

print(ff)
```

OUTPUT:

```
['This is my first line.\n', ' This is my second line \n', ' This is my third line']
```

This is my first line.

This is my second line

This is my third line

PROGRAM NO: 2

DATE:30/1/2022

AIM:Python program to copy odd lines of one file to other

INPUT:

```
f1=open("sample.txt","r")
ff=f1.readlines()

with open("odd.txt","w") as f2:

for x in range(0,len(ff)):

if(x%2!=0):

f2.write(ff[x])
```


OUTPUT:

This is my second line

PROGRAM NO: 3**DATE:30/1/2022****AIM:** Write a Python program to read each row from a given csv file and print a list of strings.**INPUT:**

```
import csv

with open('departments.csv', newline='') as csvfile:

    data = csv.reader(csvfile, delimiter=',', quotechar='"')

    for r in data:

        print(', '.join(r))
```

DEPARTMENT.CSV

```
department_id,department_name,manager_id,location_id

10,Administration,200,1700

20,Marketing,201,1800

30,Purchasing,114,1700
```

OUTPUT:

```
department_id,department_name,manager_id,location_id

10,Administration,200,1700

20,Marketing,201,1800

30,Purchasing,114,1700
```

PROGRAM NO: 4**DATE:30/1/2022****AIM:** Write a Python program to read specific columns of a given CSV file and print the content of the columns

INPUT:

```
import csv

with open('departments.csv', newline='') as csvfile:

    data = csv.DictReader(csvfile)

    print("id Department")

    for r in data:

        print(r['department_id'], " ", r['department_name'])
```

DEPARTMENT.CSV

```
department_id,department_name,manager_id,location_id

10,Administration,200,1700

20,Marketing,201,1800

30,Purchasing,114,1700
```

OUTPUT:

```
id Department

10  Administration

20  Marketing

30  Purchasing
```

PROGRAM NO: 5**DATE:30/1/2022**

AIM: Write a Python program to write a Python dictionary to a csv file. After writing the CSV file read the CSV file and display the content.

INPUT:

```
import csv

field_names = ['No', 'Company', 'Model']

cars = [
```

```
{'No': 1, 'Company': 'Ferrari', 'Model': '488 GTB'},  
{'No': 2, 'Company': 'Porsche', 'Model': '918 Spyder'},  
{'No': 3, 'Company': 'Bugatti', 'Model': 'La Voiture Noire'},  
]  
  
with open('cars.csv', 'w') as csvfile:  
  
writer = csv.DictWriter(csvfile, fieldnames=field_names)  
  
writer.writeheader()  
  
writer.writerows(cars)  
  
with open('cars.csv', newline='') as csvfile:  
  
data = csv.reader(csvfile, delimiter=' ', quotechar='|')  
  
for r in data:  
  
print(' '.join(r))
```

CARS.CSV

No,Company,Model

1,Ferrari,488 GTB

2,Porsche,918 Spyder

3,Bugatti,La Voiture Noire

OUTPUT:

No,Company,Model

1,Ferrari,488, GTB

2,Porsche,918, Spyder

3,Bugatti,La, Voiture, Noire