

## **I.COURSE OUTCOME 1(CO1)**

**Program No: 1**

**Date:24/11/2021**

**AIM :Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE**

### **IDE(Integrated development environment)**

An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger.

Eg:- NetBeans, Eclipse, IntelliJ, and Visual Studio

### **Text Editor**

A text editor is a type of computer program that edits plain text. Such programs are sometimes known as "notepad" software, following the naming of Microsoft Notepad.

Eg:- Atom, Notepad, Notepad++

### **Code analysis tool**

Code analysis tool is a method of debugging by examining source code before a program is run. It's done by analyzing a set of code against a set (or multiple sets) of coding rules. Source code analysis tools, also known as Static Application Security Testing (SAST) Tools, can help analyze source code or compiled versions of code to help find security flaws.

Eg:- PVS-Studio, Raxis, CodeSonar

## **PyCharm**

**Pycharm** is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

## **Spyder**

**Spyder** is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. ... Spyder uses Qt for its GUI and is designed to use either of the PyQt or PySide Python bindings.

## **RODEO**

**RODEO**, that is an opensource python IDE and has been brought up by the folks at yhat, is a development environment that is lightweight, intuitive and yet customizable to its very core and also contains all the features mentioned above that were searched for so long

## **Thonny**

**Thonny** is an integrated development environment for Python that is designed for beginners. It supports different ways of stepping through the code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap.

## **Atom**

**Atom** is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in JavaScript, and embedded Git Control. Its developers call it a "hackable text editor for the 21st Century", as it is fully customizable in HTML, CSS, and JavaScript.

## **Program No:2**

**Date: 24/11/2021**

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

```
a=int(input("enter the starting year="))  
b=int(input("enter the end year="))  
if(a<b):  
    print("Leap year")  
    for i in range(a,b):  
        if(i%4==0 and i%100!=0):  
            print(i,end=" ")
```

### **Output:**

enter the starting year=2021

enter the end year=2050

Leap year

2024 2028 2032 2036 2040 2044 2048

## **Program No: 3**

**Date: 24/11/2021**

**AIM :List comprehensions:**

**(a) Generate positive list of numbers from a given list of integers**

```
lt = [-10,20,35,-67,70]
r=[n for n in lt if n>=0]
print(r)
```

**Output:**

[20, 35, 70]

**(b) Square of N number**

```
n=int(input("Enter the limit"))
sqliist= [ i**2 for i in range(1,n+1)]
print("Square of N numbers : ", sqliist)
```

**Output:**

Enter the limit4

Square of N numbers : [1, 4, 9, 16]

**(c) Form a list of vowels selected from a given word**

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

**Output:**

Enter the word :Python Programming

The original string is : Python Programming

The vowel are : ['o'] ['o'] ['a'] ['i']

**(d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)**

```
word=input("Enter a word:")
```

```
print("Ordinal values corresponding to each element is:")
```

```
for i in word:
```

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

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

**Output:**

Enter a word:PYTHON

Ordinal values corresponding to each element is:

P:80 Y:89 T:84 H:72 O:79 N:78

**Program No: 4****Date: 24/11/2021****AIM :Count the occurrences of each word in a line of text.**

```
str1 = input("Enter a string : ")
word = str1.split()
count= []
for w in word:
    count.append(word.count(w))
print("count of the occurrence:" + str(list(zip(word, count))))
```

**Output:**

Enter a string : Python programming

count of the occurrence:['Python', 1]

count of the occurrence:['Python', 1), ('programming', 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**

```
n=[]
s=int(input("Enter a limit:"))
print("Enter {s} values")
for i in range(0,s):
    n.append(int(input()))
print("\nThe list after assinging:\n")
for i in range(0,len(n)):
    if(n[i]>=100):
        print("over")
    else:
        print(n[i])
```

### **Output:**

Enter a limit:2

Enter {s} values

24

199

The list after assinging:

24

Over

## **Program No: 6**

**Date: 24/11/2021**

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

```
list1=["a","d","c","a","a","a","a",]
```

```
c=list1.count("a")
```

```
print("Occurrence of a=",c)
```

### **Output:**

Occurrence of a= 5



## **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**

```
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 different 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]**

```
str2="onion"  
  
char=str2[0]  
  
str2=str2.replace(char,'$')  
  
str2=char+str2[1:]  
  
print(str2)
```

**Output:**

oni\$n

## **Program No: 9**

**Date: 24/11/2021**

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

```
str=input("Enter a string:")  
nstr=str[-1:] +str[1:-1] + str[:1]  
print("New string : ",nstr)
```

### **Output:**

Enter a string:Python

New string : nythoP

**Program No: 10****Date: 24/11/2021****AIM :Accept the radius from user and find area of circle.**

```
r=float(input("Enter the radius="))
```

```
pi=3.14
```

```
area=pi*r*r
```

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

**Output:**

```
Enter the radius=5
```

```
Area of circle= 78.5
```

## **Program No: 11**

**Date: 29/11/21**

**AIM :Find biggest of 3 numbers entered**

```
x=int(input("enter the number="))
y=int(input("enter the number="))
z=int(input("enter the number="))
if(x>y):
    if(x>z):
        print("x is large",x)
    else:
        print("z is large",z)
else:
    if(y>z):
        print("y is large",y)
    else:
        print("z is large",z)
```

### **Output:**

```
enter the number=5
enter the number=7
enter the number=2
y is large 7
```

**Program No: 12****Date: 29/11/21****AIM :Accept a file name from user and print extension of that**

```
x,y=input("Enter the file name=").split(".")
```

```
print("Extension of file=",y)
```

**Output:**

Enter the file name=program.java

Extension of file= java

## **Program No: 13**

**Date: 29/11/21**

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

```
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:green  
['red', 'blue', 'green']  
red  
green
```



## **Program No: 14**

**Date: 29/11/21**

**AIM :Accept an integer n and compute n+nn+nnn**

```
n=input("Enter the number n=")
```

```
x=int(n+n+n)
```

```
y=int(n+n)
```

```
z=int(n)
```

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

### **Output:**

```
Enter the number n=5
```

```
n+nn+nnn= 615
```

**Program No: 15****Date: 29/11/21****AIM :Print out all colors from color-list1 not contained in color-list2.**

```
list1=set(["Red","blue","green","white"])  
list2=set(["Yellow","blue","Black","white"])  
c=list1.difference(list2)  
print(c)
```

**Output:**

```
{'green', 'Red'}
```

## **Program No :16**

**Date: 29/11/21**

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

```
str1="python"  
str2="java"  
s1=str1[0]  
s2=str2[0]  
print(s2+str1[1:]+ " "+s1+str2[1:])
```

### **Output:**

jython pava

## **Program No: 17**

**Date: 29/11/21**

**AIM :Sort dictionary in ascending and descending order.**

```
import operator  
  
d={'apple':20,'orange':10,'mango':12}  
  
print("orginal dictionary")  
  
print(d)  
  
print("ascending")  
  
sdk=sorted(d.items(),key=operator.itemgetter(0))  
  
print(sdk)  
  
print("descending")  
  
sdk=sorted(d.items(),key=operator.itemgetter(0),reverse=True)  
  
print(sdk)
```

### **Output:**

```
orginal dictionary  
{'apple': 20, 'orange': 10, 'mango': 12}  
  
ascending  
[('apple', 20), ('mango', 12), ('orange', 10)]  
  
descending  
[('orange', 10), ('mango', 12), ('apple', 20)]
```

## **Program No: 18**

**Date: 29/11/21**

**AIM :Merge two dictionaries**

```
dict1={"name":'Aswin',"age":18}  
dict2={"class":'mca',"year":2021}  
print("Dictionary 1=",dict1)  
print("Dictionary 2=",dict2)  
d=dict1.copy()  
d.update(dict2)  
print("Merged dictionary",d)
```

### **Output:**

Dictionary 1= {'name': 'Aswin', 'age': 18}

Dictionary 2= {'class': 'mca', 'year': 2021}

Merged dictionary {'name': 'Aswin', 'age': 18, 'class': 'mca', 'year': 2021}

## **Program No: 19**

**Date: 29/11/21**

**AIM :Find gcd of 2 numbers.**

```
a=int(input("Enter the 1st number="))
b=int(input("Enter the 2nd number="))
i=1
gcd=0
while(i<=a and i<=b):
    if(a%i==0 and b%i==0):
        gcd=i
    i=i+1
print("GCD =",gcd)
```

### **Output:**

Enter the 1st number=120

Enter the 2nd number=5

GCD= 5

## **Program No: 20**

**Date: 29/11/21**

**AIM :From a list of integers, create a list removing even numbers.**

```
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 removing Even numbers:",num)
```

### **Output:**

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

list after removing Even numbers: [7, 25, 27]

## II. COURSE OUTCOME 2(CO2)

**Program No: 1**

**Date: 01/12/21**

**AIM :Program to find the factorial of a number**

```
n=int(input("enter the number="))  
  
f=1  
  
for i in range(1,n+1):  
    f=f*i  
  
print("Factorial of",n,"=",f)
```

**Output:**

enter the number5

Factorial of 5 = 120



## **Program No: 2**

**Date: 01/12/21**

**AIM :Generate Fibonacci series of N terms**

```
n=int(input("Enter the number="))  
  
x=0  
  
y=1  
  
s=0  
  
count=1  
  
print("Fibanocci=",end=" ")  
  
while(count<=n):  
    print(s,end=" ")  
    count+=1  
  
    x=y  
    y=s  
    s=x+y
```

### **Output:**

Enter the number=5

Fibanocci= 0 1 1 2 3

### **Program No: 3**

**Date: 01/12/21**

**AIM :Find the sum of all items in a list**

```
lst=[2,5,6,7,3]
t=sum(lst)
print("Sum of list=",t)
```

#### **Output:**

Sum of list= 23

## **Program No: 4**

**Date: 01/12/21**

**AIM :Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.**

```
from math import sqrt as s  
for i in range(1000,10000):  
    if(s(i)==int(s(i))and(i%2==0)):  
        print(i,end=" ")
```

### **Output:**

```
1024 1156 1296 1444 1600 1764 1936 2116 2304 2500 2704 2916 3136  
3364 3600 3844 4096 4356 4624 4900 5184 5476 5776 6084 6400 6724  
7056 7396 7744 8100 8464 8836 9216 9604
```

**Program No: 5****Date: 01/12/21****AIM :Display the given pyramid with step number accepted from user.**

```
n=int(input("enter the number="))  
for i in range(1,n+1):  
    for j in range(1,i+1):  
        print(i*j,end=" ")  
    print()
```

**Output:**

enter the number=3

1

2 4

3 6 9

## **Program No: 6**

**Date: 01/12/21**

**AIM :Count the number of characters (character frequency) in a string.**

```
tstr=str(input("Enter the string : "))  
  
freq = {}  
  
for i in tstr:  
    if i in freq:  
        freq[i] += 1  
    else:  
        freq[i] = 1  
  
print ("Count all characters : "+ str(freq))
```

### **Output:**

Enter the string : python

Count all characters : {'p': 1, 'y': 1, 't': 1, 'h': 1, 'o': 1, 'n': 1}

## **Program No: 7**

**Date: 08/12/21**

**AIM :Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'**

```
str=input("enter a string:")  
print("inputed string is:",str)  
if(str.endswith("ing")):  
    str=str+'ly'  
else:  
    str=str+'ing'  
print("the formated string is:",str)
```

### **Output:**

```
enter a string:programing  
inputed string is: programing  
the formated string is: programingly
```

## **Program No : 8**

**Date: 08/12/21**

**AIM :Accept a list of words and return length of longest word.**

```
a=[]  
n= int(input("Enter the number :"))  
for i in range(0,n):  
    el=input("Enter element :"+ str(i+1) )  
    a.append(el)  
    max1=len(a[0])  
    temp=a[0]  
for x in a:  
    if(len(x)>max1):  
        max1=len(i)  
        temp=i  
print("Longest Word:",temp)  
print("Length of longest word :",max1)
```

### **Output:**

Enter the number :2

Enter element 1programming

Enter element 2python

Longest Word: programming

Length of longest word : 11

## Program No: 9

**Date: 08/12/21**

**AIM :Construct following pattern using nested loop**

```
*
* *
* * *
* * * *
* * *
* *
*
```

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

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

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

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

```
    print()
```

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

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

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

```
    print()
```

**Output:**

```
*
* *
* * *
* * * *
* * *
* *
*
```



## **Program No: 10**

**Date: 08/12/21**

**AIM :Generate all factors of a number. def print\_factors(x):**

```
def factors(n):  
    print("The factors of",n,"are:")  
    for i in range(1,n+1):  
        if(n%i==0):  
            print(i)  
n=int(input("Enter a number:"))  
factors(n)
```

### **Output:**

Enter a number:16

The factors of 16 are:

1

2

4

8

16

## **Program No: 11**

**Date: 08/12/21**

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

```
import math

ta = lambda b,h:1/2*b*h

ra = lambda l,b:l*b

sa = lambda a:a*a

print("Triangle Area:",ta(10,20))

print("Rectangle Area:",ra(30,20))

print("Square Area :",sa(15))
```

### **Output:**

Triangle Area: 100.0

Rectangle Area: 600

Square Area : 225

### III. COURSE OUTCOME 3(CO3)

**Program No: 1**

**Date: 15/12/21**

**AIM :Work with built-in packages**

**Math module**

```
import math  
print("The value of pi ",math.pi)
```

```
import math as m  
print("The value of pi ",m.pi)
```

```
from math import pi,sqrt  
print(math.pi)  
print(math.sqrt(2))  
print(math.tan(90))  
print(math.cos(30))
```

**Output:**

The value of pi 3.141592653589793

The value of pi 3.141592653589793

3.141592653589793

1.4142135623730951

-1.995200412208242

0.15425144988758405

## **Time Module**

```
import time
print("current time in sec",time.time())
print("current time",time.ctime())
print("current time after 30 s",time.ctime(time.time()+30))
t=time.localtime()
print("time ",t)
print("current year",t.tm_year)
print("current month",t.tm_mon)
print("current day",t.tm_mday)
print("current hour",t.tm_hour)
print("current week",t.tm_wday)
print("day of year",t.tm_yday)
```

### **Output:**

```
current time in sec 1639933877.1339073
current time Sun Dec 19 22:41:17 2021
current time after 30 s Sun Dec 19 22:41:47 2021
time  time.struct_time(tm_year=2021, tm_mon=12, tm_mday=19,
tm_hour=22, tm_min=41, tm_sec=17, tm_wday=6, tm_yday=353,
tm_isdst=0)
current year 2021
current month 12
current day 19
current hour 22
current week 6
day of year 353
```

## Calendar module

```
import calendar

mm=int(input("Enter month"))

yy=int(input("Enter year"))

print(calendar.month(yy,mm))

print(calendar.calendar(2021))
```

## Output:

```
Enter month:11
Enter year:2021
November 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

2021

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 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7
8 9 10 11 12 13 14 8 9 10 11 12 13 14 8 9 10 11 12 13 14
15 16 17 18 19 20 21 15 16 17 18 19 20 21 15 16 17 18 19 20 21
22 23 24 25 26 27 28 22 23 24 25 26 27 28 22 23 24 25 26 27 28
29 30 31 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 1 2 1 2 3 4 5 6
5 6 7 8 9 10 11 3 4 5 6 7 8 9 7 8 9 10 11 12 13
12 13 14 15 16 17 18 10 11 12 13 14 15 16 14 15 16 17 18 19 20
19 20 21 22 23 24 25 17 18 19 20 21 22 23 21 22 23 24 25 26 27
26 27 28 29 30 24 25 26 27 28 29 30 28 29 30
31

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 1 1 2 3 4 5
5 6 7 8 9 10 11 2 3 4 5 6 7 8 6 7 8 9 10 11 12
12 13 14 15 16 17 18 9 10 11 12 13 14 15 13 14 15 16 17 18 19
19 20 21 22 23 24 25 16 17 18 19 20 21 22 20 21 22 23 24 25 26
26 27 28 29 30 31 23 24 25 26 27 28 29 27 28 29 30
30 31

October November December
Ln: 51 Col: 4
```

## **Date time module**

```
import datetime
t=datetime.time(22,56,44)
print(t)
print("hour",t.hour)
print("Minute",t.minute)
print("second",t.second)
print("Micro second",t.microsecond)
d=datetime.date.today()
print(d)
print("year",d.year)
print("month",d.month)
print("day",d.day)
d1=datetime.date.today()
print(d1)
td=datetime.timedelta(days=2)
print(td)
d2=d1+td
print(d2)
dt=datetime.datetime.combine(d1,t)
print(dt)
```

### **Output:**

22:56:44

hour 22

Minute 56

second 44

Micro second 0

2021-12-19  
year 2021  
month 12  
day 19  
2021-12-19  
2 days, 0:00:00  
2021-12-21  
2021-12-19 22:56:44

### **Statistics module**

```
import statistics as s  
print("harmonic_mean ",s.harmonic_mean([20,30,40]))  
print("mean ",s.mean([9,10,24,35,36]))  
print("median ",s.median([9,10,24,35,36]))  
print("mode ",s.mode([9,10,24,35,36]))  
print("variance ",s.variance([9,10,24,35,36]))
```

### **Output:**

```
harmonic_mean 27.69230769230769  
mean 22.8  
median 24  
mode 9  
variance 169.70000000000002
```

### **Random module**

```
import random  
random.seed(10)  
print(random.random())
```

```
print(random.uniform(20, 60))  
lst = ["orange", "apple", "graphes"]  
print(random.sample(lst, k=2))  
print(random.random())  
lst2 = ["orange", "apple", "graphes"]  
random.shuffle(lst2)  
print(lst2)  
lst3 = ["orange", "apple", "graphes"]  
print(random.choice(lst3))
```

**Output:**

```
0.5714025946899135  
37.155562187004584  
['graphes', 'orange']  
0.20609823213950174  
['orange', 'graphes', 'apple']  
Graphes
```



## **Program No: 2**

**Date: 15/12/21**

**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)**

### **Graphics**

#### **Circle**

```
def perimeter(r):  
    return(2*3.14*r)  
  
def area(r):  
    return(3.14*r*r)
```

#### **Rectangle**

```
def perimeter(l,b):  
    return(2*(l+b))  
  
def area(l,b):  
    return(l*b)
```

### **Threedgraphics**

#### **Cuboid**

```
def perimeter(l,b,h):  
    return(4*(l+b+h))  
  
def area(l,w,h):  
    return(2*l*w+2*l*h+2*h*w)
```

#### **Sphere**

```
def perimeter(r):  
    return(2*3.14*r)  
  
def area(r):
```

```
return(4*3.14*r*r)
```

## Area

```
from threedgraphics import cuboid
from threedgraphics import sphere
from graphics import rectangle
from graphics import circle
l=int(input("Enter length of cuboid:"))
w=int(input("Enter width of cuboid:"))
h=int(input("Enter height of cuboid:"))
b=int(input("Enter breadth of cuboid:"))
print("Area of cuboid=",cuboid.area(l,w,h))
print("perimeter of cuboid=",cuboid.perimeter(l,b,h))
r=int(input("Enter the radius of sphere:"))
print("Area of sphere=",sphere.area(r))
print("perimeter of sphere=",sphere.perimeter(r))
l=int(input("Enter length of rectangle:"))
b=int(input("Enter breadth of rectangle:"))
print("Area of rectangle=",rectangle.area(l,b))
print("Perimeter of rectangle=",rectangle.perimeter(l,b))
r=int(input("Enter radius of circle:"))
print("Area of Circle:",circle.area(r))
print("Perimeter of Circle:",circle.perimeter(r))
```

## Output:

Enter length of cuboid:5

Enter width of cuboid:4

Enter height of cuboid:3

Enter breadth of cuboid:7

Area of cuboid= 94

perimeter of cuboid= 60

Enter the radius of sphere:5

Area of sphere= 314.0

perimeter of sphere= 31.400000000000002

Enter length of rectangle:3

Enter breadth of rectangle:4

Area of rectangle= 12

Perimeter of rectangle= 14

Enter radius of circle:6

Area of Circle: 113.03999999999999

Perimeter of Circle: 37.68

## IV. COURSE OUTCOME 4 (CO4)

**Program No: 1**

**Date: 09/01/22**

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

```
class rectangle:
    def __init__(self,l,b):
        self.l=l
        self.b=b
    def area(self):
        area=self.l*self.b
        print("area of rectangle",area)
        return(area)
    def perimeter(self):
        per=2*(self.l+self.b)
        print("perimeter of rectangle",per)
        return(per)
r1=rectangle(4,5)
r2=rectangle(6,5)
a=r1.area()
r1.perimeter()
b=r2.area()
r2.perimeter()
if(a>b):
```

```
print("Rectangle one area is greater",a)
else:
    print("Rectangle two area is greater",b)
```

**Output:**

```
area of rectangle 20
perimeter of rectangle 18
area of rectangle 30
perimeter of rectangle 22
Rectangle two area is greater 30
```

## **Program No : 2**

**Date : 09/01/22**

**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.**

```
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 amount deposit:"))
        self.bal=self.bal+dep

    def withdraw(self):
```

```

w=int(input("Enter amount withdraw:"))
if w > self.bal:
    print("Insufficient Balance")
else:
    self.bal=self.bal-w
    print("Rs",w,"Successfully Withdrawn")

acc_no=int(input("Enter Account Number:"))
acc_name=input("Enter name:")
acc_type=input("Enter account type(savings/current):")
balance=int(input("Enter 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("Exit")
        break
    else:
        print("Invalid Option")

```

**Output:**

Enter Account Number:123

Enter name:Aswin

Enter account type(savings/current):savings

Enter initial balance:500

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:2

Enter amount deposit:500

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account info:

Account number: 123

Account name: Aswin

Account type: savings

Account balance: 1000

1.Account info



2.Deposit

3.Withdraw

4.Exit

Select your option:3

Enter amount withdraw:100

Rs 100 Successfully Withdrawn

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:1

Account info:

Account number: 123

Account name: Aswin

Account type: savings

Account balance: 900

1.Account info

2.Deposit

3.Withdraw

4.Exit

Select your option:4

Exit

## **Program No: 3**

**Date: 09/01/22**

**AIM :Create a class Rectangle with private attributes length and width.**

**Overload '<' operator to compare the area of 2 rectangles.**

```
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(" Rectangle 1")
len1=int(input("Enter length:"))
bread1=int(input("Enter breadth:"))
obj1=rectangle(len1,bread1)
obj1.area()

print(" Rectangle 2")
```

```
len2=int(input("Enter length:"))  
bread2=int(input("Enter breadth:"))  
obj2=rectangle(len2,bread2)  
obj2.area()  
if obj1 < obj2 :  
    print(" 2nd rectangle  area large:")  
else:  
    print(" 1st rectangle area large:")
```

**Output:**

Rectangle 1

Enter length:5

Enter breadth:6

Area= 30

Rectangle 2

Enter length:3

Enter breadth:8

Area= 24

1st rectangle area large:

## **Program No: 4**

**Date : 09/01/22**

**AIM :Create a class Time with private attributes hour, minute and second.**

**Overload '+' operator to find sum of 2 time.**

```
class time:
    def __init__(self,hour,minute,second):
        self.__hour=hour
        self.__minute=minute
        self.__second=second
    def __add__(self,t2):
        h=self.__hour+t2.__hour
        m=self.__minute+t2.__minute
        if(m>60):
            q=int(m/60)
            r=m%60
            h=h+q
            m=r
        s=self.__second+t2.__second
        if(s>60):
            q1=int(s/60)
            r1=s%60
            m=m+q1
            s=r1
        return(h,m,s)
print("Enter Time 1:")
h1=int(input("Hour:"))
```

```
m1=int(input("Minute:"))

s1=int(input("Second:"))
t1=time(h1,m1,s1)
print("Enter Time 2:")
h2=int(input("Hour:"))
m2=int(input("Minute:"))
s2=int(input("Second:"))
t2=time(h2,m2,s2)
h,m,s=t1+t2
print("Sum of two Times:",h,":",m,":",s)
```

### **Output**

```
Enter Time 1:
Hour:3
Minute:23
Second:22
Enter Time 2:
Hour:7
Minute:34
Second:21
Sum of two Times: 10 : 57 : 43
```

## **Program No: 5**

**Date: 09/01/22**

**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.**

```
class publisher:
    def __init__(self,title,author):
        self.title=title
        self.author=author
    def display(self):
        print("Title:",self.title)
        print("Author:",self.author)
class book(publisher):
    def __init__(self,price,no_of_page):
        self.price=price
        self.no_of_page=no_of_page
    def display(self):
        print("Price:",self.price)
        print("No. of Pages:",self.no_of_page)
class python(book):
    def __init__(self,title,author,price,no_of_page):
        publisher.__init__(self,title,author)
        book.__init__(self,price,no_of_page)
    def display(self):
```

```
print("Title:",self.title)
print("Author:",self.author)

print("Price:",self.price)
print("No. of Pages:",self.no_of_page)
p=python("Python Programming","Ashok Namdev Kamthane",500,100)
p.display()
```

**Output:**

Title: Python Programming

Author: Ashok Namdev Kamthane

Price: 500

No. of Pages: 100

## **V. COURSE OUTCOME 5 (CO5)**

**Program No: 1**

**Date : 30/01/22**

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

```
f1=open("secfile.txt","w")
f1.write("This is my first file in python.\n want to work with files \n This
is my third line")
f1=open("secfile.txt","r")
ff=f1.readlines()
print(ff)
```

**Output:**

```
['This is my first file in python.\n', ' want to work with files \n', ' This is
my third line']
```



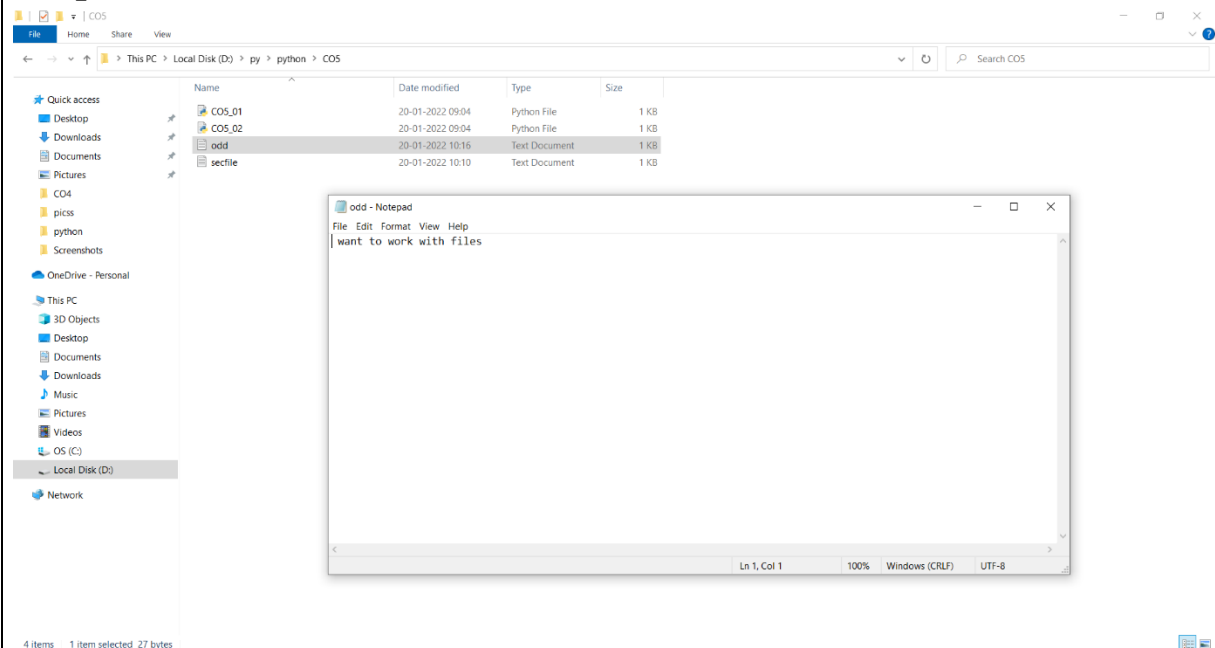
## Program No : 2

Date : 30/01/22

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

```
f1=open("secfile.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:



## **Program No: 3**

**Date : 30/01/22**

**AIM :Write a Python program to read each row from a given csv file and print a list of strings.**

```
import csv
with open('departments.csv', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=' ', quotechar='"')
    for row in data:
        print(' '.join(row))
```

### **Output:**

```
department_id,department_name,manager_id,location_id
10,Administration,200,1700
20,Marketing,201,1800
30,Purchasing,114,1700
```

### **File Content**

```
department_id,department_name,manager_id,location_id
10,Administration,200,1700
20,Marketing,201,1800
30,Purchasing,114,1700
```

## Program No: 4

Date : 30/01/22

**AIM :Write a Python program to read specific columns of a given CSV file and print the content of the columns.**

```
import csv
with open('departments.csv', newline='') as csvfile:
    data = csv.DictReader(csvfile)
    print("id Department")
    print("-----")
    for r in data:
        print(r['department_id'], " ", r['department_name'])
```

### **Output:**

id Department

-----

10 Administration

20 Marketing

30 Purchasing

## **Program No: 5**

**Date : 30/01/22**

**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.**

```
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))
```

### **Output:**

```
No,Company,Model
1,Ferrari,488, GTB
2,Porsche,918, Spyder
3,Bugatti,La, V oiture, Noire
```

#### **cars.csv**

```
No,Company,Model
1,Ferrari,488 GTB
2,Porsche,918 Spyder
3,Bugatti,La Voiture Noire
```