COURSE OUTCOME-1

DATE:18/09/2023

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

Familiarizing Integrated Development Environment (IDE), Code Analysis Tools
An integrated development environment (IDE) refers to a software application that offers computer programmers with extensive software development abilities. IDEs most often consist of a source code editor, build automation tools, and a debugger. Most modern IDEs have intelligent code completion. An IDE enables programmers to combine the different aspects of writing a computer program and increase programmer productivity by introducing features like editing source code, building executable, and debugging. IDEs are usually more feature-rich and include tools for debugging, building and deploying code. An IDE typically includes:

- A source code editor
- A compiler or interpreter
- An integrated debugger
- A graphical user interface (GUI)

A code editor is a text editor program designed specifically for editing source code. It typically includes features that help in code development, such as syntax highlighting, code completion, and debugging. The main difference between an IDE and a code editor is that an IDE has a graphical user interface (GUI) while a code editor does not. An IDE also has features such as code completion, syntax highlighting, and debugging, which are not found in a code editor. Code editors are generally simpler than IDEs, as they do not include many other IDE components. As such, code editors are typically used by experienced developers who prefer to configure their development environment manually. Some IDEs are given below:

1.IDLE

IDLE (Integrated Development and Learning Environment) is a default editor that accompanies Python. This IDE is suitable for beginner-level developers. The IDLE tool can be

used on Mac OS, Windows, and Linux. The most notable features of IDLE include:

- Ability to search for multiple files
- Interactive interpreter with syntax highlighting, and error and i/o messages
- Smart indenting, along with basic text editor features
- A very capable debugger
- A great Python IDE for Windows

2. PyCharm

PyCharm is a widely used Python IDE created by JetBrains This IDE is suitable for professional developers and facilitates the development of large Python projects.
The most notable features of PyCharm include:

- Support for JavaScript, CSS, and TypeScript
- Smart code navigation
- Quick and safe code refactoring
- Support features like accessing databases directly from the IDE

3. Visual Studio Code

Visual Studio Code (VS Code) is an open-source (and free) IDE created by Microsoft. It finds great use in Python development. VS Code is lightweight and comes with powerful features that only some of the paid IDEs offer. The most notable features of Visual Studio Code include Git integration and Code debugging within the editor.

4.SublimeText3

Sublime Text is a very popular code editor. It supports many languages, including Python. It is highly customizable and also offers fast development speeds and reliability. The most notable features of Sublime Text 3 include:

- Syntax highlighting
- Custom user commands for using the IDE
- Efficient project directory management
- It supports additional packages for the web and scientific Python development

5.Atom

Atom is an open-source code editor by GitHub and supports Python development. Atom is similar to Sublime Text and provides almost the same features emphasis on speed and usability. The most notable features of Atom include:

• Support for a large number of plugins

• Smart autocompletion

• Supports custom commands for the user to interact with the editor

• Support for cross-platform development

6. Jupyter

Jupyter is widely used in the field of data science. It is easy to use, interactive and allows live code sharing and visualization. The most notable features of Jupyter include:

• Supports for the numerical calculations and machine learning workflow

• Combine code, text, and images for greater user experience

• Intergeneration of data science libraries like NumPy, Pandas, and Matplotlib.

7. Spyder

Spyder is an open-source IDE most commonly used for scientific development. Spyder comes with Anaconda distribution, which is popular for data science and machine learning.

The most notable features of Spyder include:

• Support for automatic code completion and splitting

• Supports plotting different types of charts and data manipulation

• Integration of data science libraries like NumPy, Pandas, and Matplotlib

Code Analysis Tools

Source code analysis tools, also known as Static Application Security Testing (SAST) Tools, can help analyse source code or compiled versions of code to help find security flaws. SAST tools can be added into IDE. Such tools can help to detect issues during software development. Static code analysis techniques are used to identify potential problems in code before it is deployed, allowing developers to make changes and improve the quality of the software. Three techniques include syntax analysis, data and control flow analysis, and security analysis. SonarQube (Community Edition) is an opensource static + dynamic code analysis platform developed by Sonar Source for continuous inspection of code quality to perform fully automated code reviews / analysis to detect code smells, bugs, performance enhancements and security vulnerabilities

DATE:18/09/2023

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

PROGRAM

```
current_year=int (input ("enter current year"))
final_year=int (input ("enter final year"))
print ("leap year from current year to final year is")
for year in range (current_year, final_year):
  if((year%4==0) and (year%100! =0)) or(year%400==0):
    print(year)
```

OUTPUT

enter current year:2000
enter final year:2025
leap year from current year to final year is
2000
2004
2008
2012
2016
2020
2024

DATE: 18/09/2023

3. List comprehensions:

a. **Generate positive list of numbers from a given list of integers**
b. **Square of N numbers**
c. **Form a list of vowels selected from a given word**
d. **List ordinal value of each element of a word (Hint: use ord() to get ordinal values)**

PROGRAM

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

```
list1=[1, -5, -12,5,4,10]
list2=[i for i in list1 if i>0]
print ("positive numbers are", list2)
```

OUTPUT

positive numbers are [1, 5, 4, 10]

b. Square of N numbers

OUTPUT

```
list1=[2,3,4,5,7]
list2=[i**2 for i in list1]
print ("the square is",list2)
```

c. Form a list of vowels selected from a given word

OUTPUT

```
a=input ("Enter a word:")
list1= [i for i in a if i in 'aeiouAEIOU']
list1
```

d.List ordinal value of each element of a word

```
a=input("Enter a char:")
b=[ord(i) for i in a]
b
```

OUTPUT

Enter a char: g
[103]

DATE: 18/09/2023

4. Count the occurrences of each word in a line of text.

PROGRAM

```
n="It is a beautiful Day and a sunny day too"
a=input ("Enter the word:")
print(n.count(a))
```

OUTPUT

Enter the word: beautiful
1

Enter the word: summer
0

DATE:18/09/2023

5. Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

PROGRAM

```
n=int (input ("Enter the  number  of integers:"))
list= []
for i in range(n):
   a=int (input ("Enter the elements"))
   if a>100:
     list.append("over")
   else:
      list.append(a)
print(list)
```

OUTPUT

```
Enter the number of integers:3
Enter the elements103
Enter the elements55
Enter the elements208
['over', 55, 'over']
```

DATE:20/09/2023

6. Store a list of first names. Count the occurrences of ‘a’ within the list

PROGRAM

```
name=['Arun','Anandhu','Meenakshy','Jerry']
for i in name:
 print(" a occurs in",i,i.count('a'),"times")
```

OUTPUT

```
a occurs in Arun 1 times
a occurs in Anandhu 2 times
a occurs in Meenakshy 1 times
a occurs in Jerry 0 times
```

DATE:20/09/2023

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

PROGRAM

```
import numpy  as np
l1=[1,2,3,4,5]
l2=[2,4,8,9,1,6]
print("Length of l1 is:",len (l1))
print ("Length of l2 is:",len (l2))
if len(l1)==len(l2):
  print("equal length")
else:
  print("l1 and l2 are not equal length")
  sum1=0
  sum2=0
for i in  l1:
  sum1=i+sum1
print("sum of l1 is:",sum1)
for i in l2:
  sum2=i+sum2
print("sum of l2 is:",sum2)
if sum(l1)==sum(l2):
  print("equal sum")
else:
  print("sum is not equal")
l1a=np.array(l1)
l2a=np.array(l2)
list=[]
for i in l1a:
 for j in l2a:
  if i==j:
    list.append(i)
print(list)
```

OUTPUT

Length of l1 is: 5
Length of l2 is: 6
l1 and l2 are not equal length
sum of l1 is: 15
sum of l2 is: 30
sum is not equal
[1, 2, 4]

DATE:20/09/2023

8.Get a string from an input string where all occurrences of first character replaced with '$', except first character.

PROGRAM

```
n=(input("Enter a string:"))
c=n [0]
n=n.replace(c,"$")
n=c+n[1:]
print(n)
```

OUTPUT

Enter a string: onion
oni$n

DATE: 20/09/2023

9.Create a string from given string where first and last characters exchanged.

PROGRAM

```
name="python"
name[-1]+name[1:-1]+name[0]
```

OUTPUT

nythop

DATE:27/9/2023

10.Accept the radius from user and find area of circle.

PROGRAM

```
r=float (input ("Enter the radius of the circle:"))
area=3.14*r*r
print("Area of circle is:", area)
```

OUTPUT

Enter the radius of the circle:4
Area of circle is: 50.24

DATE: 27/9/2023

11.Find biggest of 3 numbers entered.

PROGRAM

```
a=int(input("Enter value of a:"))
b=int(input("Enter value of b:"))
c=int(input ("Enter value of c:"))
if(a==b==c):
  print( "a,b and c are equal")
elif(a>b and a>c):
  print("a is largest")
elif(b>a and b>c):
  print("b is largest")
else:
  print(c," is largest")
```

OUTPUT

```
Enter value of a:23
Enter value of b:12
Enter value of c:13
a is largest

Enter value of a:23
Enter value of b:45
Enter value of c:12
b is largest

Enter value of a:13
Enter value of b:19
Enter value of c:34
c is largest
```

Enter value of a:13
Enter value of b:13
Enter value of c:13
a, b and c are equal

DATE:27/9/2023

12.Accept a file name from user and print extension of that.

PROGRAM

```
name=input("Enter a file name :")
file_extension=name.split(".")[-1]
print("Extension of the file is:",file_extension)
```

OUTPUT

```
Enter a file name :file.html
Extension of the file is: html
```

DATE:27/03/2023

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

PROGRAM

```
n=int(input("Enter the number of colors:"))
l=[]
for i in range(n):
  color=input("Enter the colors:")
  l.append(color)
print(l)
print(l[0])
print(l[-1])
```

OUTPUT

```
Enter the number of colors:3
Enter the colors:red
Enter the colors:yellow
Enter the colors:green
['red', 'yellow', 'green']
red
green
```

DATE:4/10/2023

14.Accept an integer n and compute n+nn+nnn.

PROGRAM

```
n=int(input("enter an integer:"))
nn=11*n
nnn=111*n
res=n+nn+nnn
print("result is",res)
```

OUTPUT

```
enter an integer:5
result is 615
```

DATE: 4/10/2023

15.Print out all colors from color-list1 not contained in color-list2.

PROGRAM

```
list1 = ["Purple", "Black”, “Green", "Blue", "Orange"]
list2 = ["Green", "Yellow", "Purple","Grey"]
result = []
for color in list1:
    if color not in list2:
        result.append(color)
print("Colors in list1 not in list2:")
for color in result:
    print(color)
```

OUTPUT

Colors in list1 not in list2:
Black
Blue
Orange

DATE: 4/10/2023

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

PROGRAM

```
string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")
if len(string1) > 1 and len(string2) > 1:
    swapped_string = string2[0] + string1[1:] + " " + string1[0] + string2[1:]
    print("Result:", swapped_string)
else:
    print("Strings are  short to swap at position 1.")
```

OUTPUT

Enter the first string: Beautiful
Enter the second string: Girl
Result: Geautiful Birl

DATE: 4/10/2023

17. Sort dictionary in ascending and descending order.

PROGRAM

```
dict = {'yellow':1,'orange': 2,'red':3,'green':6}
dict_asc = {}
for key in sorted(dict):
    dict_asc[key] = dict[key]
dict_desc = {}
for key in reversed(sorted(dict)):
   dict_desc[key] = dict[key]
print("Dictionary sorted in ascending order:")
print(dict_asc)
print("\nDictionary sorted in descending order:")
print(dict_desc)
```

OUTPUT

Dictionary sorted in ascending order:
{'green': 6, 'orange': 2, 'red': 3, 'yellow': 1}

Dictionary sorted in descending order:
{'yellow': 1, 'red': 3, 'orange': 2, 'green': 6}

DATE:9/10/2023

18. Merge two dictionaries.

PROGRAM

```
dict1 = {'Anu': 1, 'Bhama': 2}
dict2 = {'Anandhu': 3, 'Meenakshy': 4}
dict1.update(dict2)
print("Merged dictionary is:", dict1)
```

OUTPUT

Merged dictionary is: {'Anu': 1, 'Bhama': 2, 'Anandhu': 3, 'Meenakshy': 4}

DATE: 9/10/2023

19. Find gcd of 2 numbers.

PROGRAM

```
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
gcd = 1
for i in range(1, min(num1, num2) + 1):
    if num1 % i == 0 and num2 % i == 0:
        gcd = i
print(f"The GCD of {num1} and {num2} is: {gcd}")
```

OUTPUT

```
Enter the first number: 12
Enter the second number: 24
The GCD of 12 and 24 is: 12
```

DATE: 9/10/2023

20.From a list of integers, create a list removing even numbers.

PROGRAM

```
list1 = []
list2 = []
n = int(input("Enter n:"))
while n > 0:
    num = int(input("Enter number:"))
    list1.append(num)
    n -= 1
print("List before removing even numbers:", list1)
list2 = [i for i in list1 if i % 2 != 0]
print("List after removing even numbers:", list2)
```

OUTPUT

```
Enter n:4
Enter number:4
Enter number:3
Enter number:8
Enter number:7
List before removing even numbers: [4, 3, 8, 7]
List after removing even numbers: [3, 7]
```

COURSE OUTCOME-2

DATE:9/10/2023

1.Program to find the factorial of a number

PROGRAM

```
n=int(input(“Enter a number:”))
i=1
fact=1
while(i<=n):
  fact=fact*i
  i=i+1
print(“factorial is”,fact)
```

OUTPUT

Enter a number:5
factorial is 120

DATE:11/10/2023

2.Generate Fibonacci series of N terms

PROGRAM

```
t1=0
t2=1
n=int(input("Enter a limit:"))
print("fibonacci series is:")
print(t1)
print(t2)
for i in range(3,n+1):
  print
  next=t1+t2
  print(next)
  t1=t2
  t2=next
```

OUTPUT

```
Enter a limit:5
fibonacci series is:
0
1
1
2
3
```

DATE: 11/10/2023

3. Find the sum of all items in a list

PROGRAM

```
list=[1,4,3,7,8]
total=0
for i in list:
  total=total+i
print("sum of all items in the list is:",total)
```

OUTPUT

sum of all items in the list is: 23

DATE: 11/10/2023

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

PROGRAM

```
res = []
for num in range(1000, 10000):
    if all(int(digit) % 2 == 0 for digit in str(num)):
        sqrt = int(num**0.5)
        if sqrt*sqrt == num:
            res.append(num)
print("List of four-digit even and perfect square numbers:", res)
```

OUTPUT

List of four-digit even, perfect square numbers: [4624, 6084, 6400, 8464]

DATE: 11/10/2023

5.Display the given pyramid with step number accepted from user. Eg: N=4

1

2 4

3 6 9

4 8 12 16

PROGRAM

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

OUTPUT

```
Enter number of rows:4
1
2 4
3 6 9
4 8 12 16
```

DATE:16/10/2023

6.Count the number of characters (character frequency) in a string

PROGRAM

```
string = input("Enter a string: ")
char_frequency = {}
for char in string:
    if char in char_frequency:
        char_frequency[char] += 1
    else:
        char_frequency[char] = 1
for char, count in char_frequency.items():
    print(f"'{char}': {count} times")
```

OUTPUT

```
Enter a string: Aeroplane
'A': 1 times
'e': 2 times
'r': 1 times
'o': 1 times
'p': 1 times
'l': 1 times
'a': 1 times
'n': 1 times
```

DATE: 16/10/2023

7.Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’

PROGRAM

```
input_string = input("Enter a string: ")
if input_string.endswith("ing"):
   result_string = input_string + "ly"
else:
   result_string = input_string + "ing"
print("Modified string:", result_string)
```

OUTPUT

Enter a string: morning
Modified string: morningly

Enter a string: Rain
Modified string: Raining

DATE: 16/10/2023

8.Accept a list of words and return length of longest word.

PROGRAM

```
num_words = int(input("Enter number of words:"))
words = []
for i in range(num_words):
    word = input("Enter a word: ")
    words.append(word)
longest_length = max(len(word) for word in words)
print("Length of the longest word:", longest_length)
```

OUTPUT

```
Enter number of words:3
Enter a word: Mathematics
Enter a word: Statistics
Enter a word: Literature
Length of the longest word: 11
```

DATE: 16/10/2023

9.Construct following pattern using nested loop

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

PROGRAM

```
from numpy.lib.function_base import i0
n=int(input("Enter value of n :"))
for i in range(1,n+1):
  print("*"*i)
for i in range(n-1,0,-1):
  print("*"*i)
```

OUTPUT

```
Enter value of n :5
*
**
***
****
*****
****
***
**
*
```

DATE:18/10/2023

10. Generate all factors of a number.

PROGRAM

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

OUTPUT

```
Enter a number:12
The factors are:
1
2
3
4
6
12
```

DATE:18/10/2023

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

PROGRAM

```
s=float(input("Enter the side of the square:"))
square_area=lambda s:s*s
print("area of square is:",s*s)
l=float(input("Enter length of rectangle:"))
b=float(input("Enter breadth  of rectangle:"))
rect_area=lambda l,b:l*b;
print("area of rectangle is:",l*b)
x=float(input("enter the base of triangle:"))
y=float(input("enter the height of triangle:"))
triangle_area=lambda x,y: 0.5*x*y
print("area of triangle is:",0.5*x*y)
```

OUTPUT

```
Enter the side of the square:4
area of square is: 16.0
Enter length of rectangle:5
Enter breadth  of rectangle:8
area of rectangle is: 40.0
enter the base of triangle:10
enter the height of triangle:5
area of triangle is: 25.0
```

COURSE OUTCOME-5

DATE:16/10/2023

1.Write a python program to read a file line by line and store it into a list.

PROGRAM

```
with open("stud.txt") as f:
  slist=f.readlines()
print(slist)
slist=[x.strip() for x in slist]
print("The contents of the file are:")
print(slist)
```

OUTPUT

The contents of the file are:
['Hello How are you', 'Good Morning', 'have a nice day', 'Have a beautiful day', 'Its raining']

DATE:16/10/2023

2.Python program to copy odd lines of the file stud.txt to odd.txt and copy the even lines of the files to even.txt.

PROGRAM

stud.txt

```
Hello How are you
Good Morning
have a nice day
Have a beautiful day
Its raining
```

sfile.py

```
sfile=open("stud.txt","r")
ofile=open("odd.txt","w")
efile=open("even.txt","w")
content=sfile.readlines()
print("content of the file are:")
print(content)
for i in range(len(content)):
 if(i%2==0):
   efile.write(content[i])
 else:
   ofile.write(content[i])
s.close()
o.close()
e.close()
```

OUTPUT

odd.txt

Good Morning
Have a beautiful day

even.txt

Hello How are you
have a nice day
Its raining.

DATE:21/10/2023

3.Write a python program to read each row from a given csv file and print a list of strings.

PROGRAM

```
import csv
with open("Data.csv",'r') as f:
  data=csv.reader(f)
  for i in data:
  print(i)
```

OUTPUT

```
['Rollno', 'Name', 'Age']
['1', 'Arun', '20']
['2', 'Anju', '18']
['3', 'Amal, '23']
['4', 'Manu', '22']
['5', 'Ram', '29']
```

DATE:21/10/2023

4.Write a python program to read specific columns of a given CSV file and print the contents of the column.

PROGRAM

```
import csv
n=int(input("Enter the line number : "))
with open("Data.csv",'r') as f:
    data=list(csv.reader(f))
    print(data[n])
```

OUTPUT

```
Enter the line number : 2
['1', 'Meenakshy', '46']
```

DATE:21/10/2023

5.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.

PROGRAM

```
import csv
import pandas
field=['name','Rollno','Age']
sdict=[{'Name':'ram','Rollno':23,'Age':34},
       {'name':'anu','Rollno':28,'Age':39}
       {'name':'arun','Rollno':48,'Age':39]
with open("dpt.csv","w") as dfile:
  writer=csv.DictWriter(dfile,fieldnames=field)
  writer.writeheader()
  writer.writerows(sdict)
data=pandas.read_csv("dpt.csv")
print(data)
```

OUTPUT

	Name	Rollno	Age
0	ram	23	34
1	anu	28	39
2	arun	48	39

COURSE OUTCOME 3

Date:15/11/2023

1. Work with built-in packages.

BUILT-IN PACKAGES IN PYTHON

Python comes with a comprehensive standard library that includes a wide range of built-in packages and modules. These modules provide functionality for tasks ranging from file I/O to web development. Here are some commonly used built-in packages in Python:

1. os : Operating system interface, provides a way of using operating system-dependent functionality like reading or writing to the file system.

```
import os
```

2. sys : Provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter.

```
import sys
```

3. math : Mathematical functions such as basic arithmetic operations, logarithms, trigonometric functions, etc.

```
import math
```

4. datetime : Date and time handling.

```
import datetime
```

5. json : JSON encoder and decoder.

```
import json
```

6. urllib : URL handling modules, including parsing, quoting, and fetching.

```
from urllib import request, parse
```

7. random : Generate pseudo-random numbers.

```
import random
```

8. re : Regular expression operations.

    import re

9. collections : Implements specialized container datatypes.

    from collections import Counter, defaultdict

10. sqlite3 : SQLite database interface.

    import sqlite3

11. csv : CSV file reading and writing.

    import csv

12. gzip : Support for gzip files.

    import gzip

13. socket : Low-level networking interface.

    import socket

14. argparse : Command-line argument parsing.

    import argparse

COURSE OUTCOME 3

Date: 15/11/2023

2. 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)

Program

```
# packages/graphics/rectangle.py
def area (length,width):
        return length*width
def perimeter (length,width):
        return 2*(length+width)

# packages/graphics/circle.py
import math
def area(radius):
        return math.pi*radius*radius
def perimeter(radius):
        return 2 * math.pi * radius

# packages/graphics/threedgraphics/cuboid.py
def surfacearea(length, width, height):
        return 2 * (length * width + width * height + height * length)
def volume(length, width, height):
        return length * width * height

# packages/graphics/threedgraphics/sphere.py
import math
def surface_area(radius):
        return 4 * math.pi * radius**2
def volume(radius):
        return (4/3) * math.pi * radius**3

# packages/main.py
from graphics import rectangle,circle
from graphics.threedgraphics import cuboid,sphere
```

```
# Using rectangle module
length =int(input("Enter length: "))
width =int(input("Enter width: "))
print("Area of rectangle: ",rectangle.area(length,width))
print("Perimeter of rectangle: ",rectangle.perimeter(length,width))
# Using circle module
radius=int(input("Enter radius: "))
print("Area of circle: ",circle.area(radius))
print("Perimeter of circle: ",circle.perimeter(radius))
# Using cuboid module from 3Dgraphics sub-package
cuboid_length =int(input("Enter length: "))
cuboid_width = int(input("Enter width: "))
cuboid_height = int(input("Enter height: "))
print("Cuboid Surface Area: ", cuboid.surfacearea(cuboid_length, cuboid_width,
cuboid_height))
print("Cuboid volume: ", cuboid.volume(cuboid_length, cuboid_width,
cuboid_height))
# Using sphere module from 3Dgraphics sub-package
sphere_radius = int(input("Enter radius: "))
print("Sphere Surface Area: ", sphere.surface_area(sphere_radius))
print("Sphere Volume: ", sphere.volume(sphere_radius))
```

Output

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~$ python3 main.py
Enter length: 2
Enter width: 4
Area of rectangle:8
Perimeter of rectangle:12
Enter radius: 4
Area of circle: 50.26548245743669
Perimeter of circle:  25.132741228718345
Enter length: 3
Enter width: 4
Enter height: 5
Cuboid Surface Area:  94
Cuboid volume:  60
Enter radius: 4
Sphere Surface Area: 201.06192982974676
Sphere Volume: 268.082573106329
mits@mits-HP-280-Pro-G6-Microtower-PC:~$
```

COURSE OUTCOME 4

DATE: 27/11/2023

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

PROGRAM

```
class Rectangle:
  def__init_(self,length,breadth):
     self.length=length
     self.breadth=breadth
  def area(self):
     return self.length*self.breadth
  def perimeter(self):
     return 2*(self.length+self.breadth)
  def comparison_area(self,rect):
          if(self.area()>rect.area()):
                  return "rectangle 1 is larger"
          elif(self.area()<rect.area()):
                  return "rectangle 2 is larger"
          else:
                  return "both are equal"
print("first rectangle")
length=int(input("enter the length:"))
breadth=int(input("enter the breadth:"))
rectangle1=Rectangle(length,breadth)
print("area of rectangle1:",rectangle1.area())
print("perimeter of rectangle1:",rectangle1.perimeter())
print("\nsecond rectangle")
length=int(input("enter the length:"))
breadth=int(input("enter the breadth:"))
```

```
rectangle2=Rectangle(length,breadth)

print("area of rectangle2:",rectangle2.area())

print("perimeter of rectangle2:",rectangle2.perimeter())

result=rectangle1.comparison_area(rectangle2)

print(result)
```

OUTPUT

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/s1mca45 $ python3 rect.py
first rectangle
enter the length:3
enter the breadth:3
area of rectangle1: 9
perimeter of rectangle1: 12

second rectangle
enter the length:4
enter the breadth:3
area of rectangle2: 12
perimeter of rectangle2:14
rectangle 2 is larger
```

DATE: 27/11/2023

2. 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.

PROGRAM

```
class Bankaccount:
  def__init_(self,accno,name,acctype,balance=0):
    self.accno=accno
    self.name=name
    self.acctype=acctype
    self.balance=balance
  def deposit(self,amount):
    if(amount>0):
        self.balance=self.balance+amount
        print("successfull deposit of",amount)
        print("new balance:",self.balance)
    else:
        print("not successfull")
  def withdraw(self,amount):
    if(0<amount<self.balance):
       self.balance=self.balance-amount
    elif(amount>self.balance):
       print("not possible to withdraw")
    else:
       print("invalid")
  def getbalance(self):
     print("current balance:",self.balance)
accno=int(input("enter the account number:"))
name=input("enter your name:")
acctype=input("enter the type of account:")
account1=Bankaccount(accno,name,acctype)
while True:
  print("1.deposit \n 2.withdraw \n 3.balance \n 4.exit")
  ch=int(input("enter your choice:"))
  if(ch==1):
     damount=int(input("enter the amount to be deposited:"))
     account1.deposit(damount)
  elif(ch==2):
     wamount=int(input("enter the amount to be withdraw:"))
     account1.withdraw(wamount)
  elif(ch==3):
     account1.getbalance()
  elif(ch==4):
     exit(0)
  else:
     print("wrong choice")
```

OUTPUT

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/s1mca45 $ python3 bankacc.py
enter the account number:101
enter your name:Ram
enter the type of account:savings
***MENUDRIVEN***
1.deposit
 2.withdraw
 3.balance
 4.exit
enter your choice:1
enter the amount to be deposited:2000
successfull deposit of 2000
new balance: 2000
***MENUDRIVEN***
1.deposit
 2.withdraw
 3.balance
 4.exit
enter your choice:3
current balance: 2000
***MENUDRIVEN***
1.deposit
 2.withdraw
 3.balance
 4.exit
enter your choice:2
enter the amount to be withdraw:1000
***MENUDRIVEN***
1.deposit
 2.withdraw
 3.balance
 4.exit
enter your choice:3
current balance: 1000
***MENUDRIVEN***
1.deposit
 2.withdraw
 3.balance
 4.exit
enter your choice:5
wrong choice
***MENUDRIVEN***
1.deposit
 2.withdraw
 3.balance
 4.exit
enter your choice:4
```

DATE: 29/11/2023

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

PROGRAM

```
class Rectangle:
  def__init_(self,length,breadth):
      self.length=length
      self.breadth=breadth
  def area(self):
    return self.length * self.breadth
  def__gt_(self, other):
      if(self.area()>other.area()):
         return True
      else:
         return False
l1=int(input("Enter the length of first rectangle:"))
b1=int(input("Enter the breadth of first rectangle:"))
l2=int(input("Enter the length of second rectangle:"))
b2=int(input("Enter the breadth of second rectangle:"))
ob1 = Rectangle(l1,b1)
ob2 = Rectangle(l2,b2)
if(ob1>ob2):
   print("ob1 is greater than ob2")
else:
   print("ob2 is greater than ob1")
```

OUTPUT

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/s1mca45$ python3 rectangle.py
Enter the length of first rectangle:4
Enter the breadth of first rectangle:2
Enter the length of second rectangle:7
Enter the breadth of second rectangle:5
ob2 is greater than ob1
```

DATE: 29/11/2023

4. Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

PROGRAM

```
class Time:
  def__init_(self, hour=0, minute=0, second=0):
    self._hour = hour # Private attribute
  self._minute = minute # Private attribute
  self._second = second # Private attribute
  def__add_(self, other):
    total_seconds = self._hour * 3600 + self._minute * 60 + self._second + \
    other._hour * 3600 + other._minute * 60 + other._second
    new_hour, remainder = divmod(total_seconds, 3600)
  new_minute, new_second = divmod(remainder, 60)
  return Time(new_hour, new_minute,  new_second)
  def__str_(self):
     return f"{self._hour:02d}:{self._minute:02d}:{self._second:02d}"
 h1=int(input("Enter the hour"))
m1=int(input("Enter the minute:"))
s1=int(input("Enter the second:"))
time1=Time(h1,m1,s1)
h2=int(input("Enter the hour:"))
m2=int(input("Enter the minute:"))
s2=int(input("Enter the second:"))
time2=Time(h2,m2,s2)
 sum_time = time1 + time2
 print("Time 1:", time1)
 print("Time 2:", time2)
 print("Sum of Time 1 and Time 2:", sum_time)
```

OUTPUT

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/s1mca45$ python3 time.py
Enter the hour:5
Enter the minute:3
Enter the second:2
Enter the hour:3
Enter the minute:45
Enter the second:32
Time 1: 05:03:02
Time 2: 03:45:32
Sum of Time 1 and Time 2: 08:48:34
```

DATE: 29/11/2023

5. 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.

PROGRAM

```
class Publisher:
  def__init__(self, name):
    self.name = name
class Book(Publisher):
  def__init__(self, name, title, author):
    super()._init_(name)
    self.title = title
    self.author = author
  def display_info(self):
    print("Publisher:", self.name)
    print("Title:", self.title)
    print("Author:", self.author)
class Python(Book):
  def__init_(self, name, title, author, price, no_of_pages):
    super()._init_(name, title, author)
    self.price = price
    self.no_of_pages = no_of_pages
  def display_info(self): # Method overriding
    super().display_info() # Invoking the base class (Book) method
    print("Price:", self.price)
    print("Number of Pages:", self.no_of_pages)
name_p=input("Enter the name of publisher:")
title_b=input("Enter the title of book:")
name_a=input("Enter the name of author:")
```

```
price=int(input("Enter the price of book:"))

noofpages=int(input("Enter the no of pages of book:"))

python_book = Python(name_p,title_b,name_a,price,noofpages )

python_book.display_info()
```

OUTPUT

```
mits@mits-HP-280-Pro-G6-Microtower-PC:~/Desktop/s1mca45 $ python3 publisher.py
Enter the name of publisher: Ram Gopal
Enter the title of book: Wind
Enter the name of author:John Soy
Enter the price of book:500
Enter the no of pages of book:400
Publisher: Ram Gopal
Title: Wind
Author: John Soy
Price: 500
Number of Pages: 400
```

COURSE OUTCOME-5

DATE:16/10/2023

1.Write a python program to read a file line by line and store it into a list.

PROGRAM

```
with open("stud.txt") as f:
  slist=f.readlines()
print(slist)
slist=[x.strip() for x in slist]
print("The contents of the file are:")
print(slist)
```

OUTPUT

The contents of the file are:
['Hello How are you', 'Good Morning', 'have a nice day', 'Have a beautiful day', 'Its raining']

DATE:16/10/2023

2.Python program to copy odd lines of the file stud.txt to odd.txt and copy the even lines of the files to even.txt.

PROGRAM

stud.txt

```
Hello How are you
Good Morning
have a nice day
Have a beautiful day
Its raining
```

sfile.py

```
sfile=open("stud.txt","r")
ofile=open("odd.txt","w")
efile=open("even.txt","w")
content=sfile.readlines()
print("content of the file are:")
print(content)
for i in range(len(content)):
 if(i%2==0):
   efile.write(content[i])
 else:
   ofile.write(content[i])
s.close()
o.close()
e.close()
```

OUTPUT

odd.txt

Good Morning
Have a beautiful day

even.txt

Hello How are you
have a nice day
Its raining.

DATE:21/10/2023

3.Write a python program to read each row from a given csv file and print a list of strings.

PROGRAM

```
import csv
with open("Data.csv",'r') as f:
  data=csv.reader(f)
  for i in data:
  print(i)
```

OUTPUT

```
['Rollno', 'Name', 'Age']
['1', 'Arun', '20']
['2', 'Anju', '18']
['3', 'Amal, '23']
['4', 'Manu', '22']
['5', 'Ram', '29']
```

DATE:21/10/2023

4.Write a python program to read specific columns of a given CSV file and print the contents of the column.

PROGRAM

```
import csv
n=int(input("Enter the line number : "))
with open("Data.csv",'r') as f:
    data=list(csv.reader(f))
    print(data[n])
```

OUTPUT

```
Enter the line number : 2
['1', 'Meenakshy', '46']
```

DATE:21/10/2023

5.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.

PROGRAM

```
import csv
import pandas
field=['name','Rollno','Age']
sdict=[{'Name':'ram','Rollno':23,'Age':34},
       {'name':'anu','Rollno':28,'Age':39}
       {'name':'arun','Rollno':48,'Age':39]
with open("dpt.csv","w") as dfile:
  writer=csv.DictWriter(dfile,fieldnames=field)
  writer.writeheader()
  writer.writerows(sdict)
data=pandas.read_csv("dpt.csv")
print(data)
```

OUTPUT

	Name	Rollno	Age
0	ram	23	34
1	anu	28	39
2	arun	48	39