

COURSE OUTCOME 1

Date: 18/09/2023

1. 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. Sublime Text 3

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 with 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 open source static + dynamic code analysis platform developed by SonarSource for continuous inspection of code quality to perform fully automated code reviews / analysis to detect code smells, bugs, performance enhancements and security vulnerabilities.

COURSE OUTCOME 1

Date: 18/09/2023

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

Program

```
c=int(input("Enter the current year "))
n=int(input("Enter the final year"))
for i in range(c,n):
    if(i%4==0) and (i%100!=0) or (i%400==0):
        print(i)
```

Output

```
Enter the current year 2023
Enter the final year2050
2024
2028
2032
2036
2040
2044
2048
```

COURSE OUTCOME 1

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)

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

Program

```
list1=[2,5,-3,4,-9,7,6]
list2=[i for i in list1 if i>0]
print("positive numbers",list2)
```

Output

```
positive numbers [2, 5, 4, 7, 6]
```

b. Square of N numbers

Program

```
list1=[1,-2,3,-4,5]
list2=[i**2 for i in list1]
print("square of numbers",list2)
```

Output

```
square of numbers [1, 4, 9, 16, 25]
```

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

Program

```
w=input("Enter the word : ")
l=[i for i in w if i in"aeiouAEIOU" ]
print(l)
```

Output

```
Enter the word : paint
['a', 'i']
```

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

Program

```
c=input("Enter the character : ")
l=[ord(i) for i in c]
print(l)
```

Output

```
Enter the character : abcd
[97, 98, 99, 100]
```

COURSE OUTCOME 1

Date: 18/09/2023

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

Program

```
n="betty bought some butter but the butter was bitter betty bought some  
better butter to make the bitter butter better"  
w=input("Enter a word : ")  
print(n.count(w))
```

Output

```
Enter a word : butter  
4
```

COURSE OUTCOME 1

Date: 18/09/2023

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

Program

```
list=[]
n=int(input("Enter number of elements : "))
for i in range(0,n):
    e=int(input("Enter elements : "))
    if (e>100):
        list.append("over")
    else:
        list.append(e)
print(list)
```

Output

```
Enter number of elements : 3
Enter elements : 52
Enter elements : 105
Enter elements : 100
[52, 'over', 100]
```


COURSE OUTCOME 1

Date: 20/09/2023

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

Program

```
n=['rachel','rishon','jessy','jolly']  
for i in n:  
    print("a occurs in",i,i.count('a'),"times")
```

Output

```
a occurs in rachel 1 times  
a occurs in rishon 0 times  
a occurs in jessy 0 times  
a occurs in jolly 0 times
```

COURSE OUTCOME 1

Date: 20/09/2023

7. 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

Program

```
list1=[]
n=int(input("Enter number of elements in first list: "))
for i in range(0,n):
    e=int(input("Enter elements in first list : "))
    list1.append(e)
list2=[]
n2=int(input("Enter number of elements in second list : "))
for i in range(0,n2):
    e2=int(input("Enter elements in second list : "))
    list2.append(e2)
if (len(list1)==len(list2)):
    print("The lists are of same length")
else:
    print("The lists are of different length")
if(sum(list1)==sum(list2)):
    print("The lists sums to same value")
else:
    print("The lists does not sum to same value")
    print("Values that occur in both lists are")
    for i in list1:
        for j in list2:
            if(i==j):
                print(i)
```

Output

```
Enter number of elements in first list: 1
Enter elements in first list : 1
Enter number of elements in second list : 3
Enter elements in second list : 1
Enter elements in second list : 2
Enter elements in second list : 3
The lists are of different length
The lists does not sum to same value
Values that occur in both lists are
1
```

COURSE OUTCOME 1

Date: 20/09/2023

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

Program

```
s=input("Enter string : ")
s1=s[0]+s[1:].replace(s[0],'$')
print(s1)
```

Output

```
Enter string : remember
remembe$
```

COURSE OUTCOME 1

Date: 20/09/2023

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

Program

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

Output

lacher

COURSE OUTCOME 1

Date: 20/09/2023

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

Program

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

Output

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

COURSE OUTCOME 1

Date: 27/09/2023

11. Find biggest of 3 numbers entered.

Program

```
a=int(input("Enter the first number : "))
b=int(input("Enter the second number : "))
c=int(input("Enter the third number : "))
if(a>b and a>c):
    print("The biggest number is : ",a)
elif(b>a and b>c):
    print("The biggest number is : ",b)
elif(c>a and c>b):
    print("The biggest number is : ",c)
```

Output

```
Enter the first number : 5
Enter the second number : 6
Enter the third number : 8
The biggest number is : 8
```

COURSE OUTCOME 1

Date: 27/09/2023

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

Program

```
f=input("Enter file name : ")
r=f.split(".")
print(r[1])
```

Output

```
Enter file name : file.html
html
```

COURSE OUTCOME 1

Date: 27/09/2023

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

Program

```
list=[]
n=int(input("Enter number of colours in the list: "))
for i in range(0,n):
    e=input()
    list.append(e)
print("The list of colours : ",list)
print("The first colour in the list is ",list[0])
print("The last colour in the list is ",list[-1])
```

Output

```
Enter number of colours in the list: 4
Black
Indigo
Green
White
The list of colours :  ['Black', 'Indigo', 'Green', 'White']
The first colour in the list is  Black
The last colour in the list is  White
```


COURSE OUTCOME 1

Date: 27/09/2023

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

Program

```
n=int(input("Input an integer : "))
n1=n
n2=n*n
n3=n*n*n
print(n1+n2+n3)
```

Output

```
Input an integer : 4
84
```

COURSE OUTCOME 1

Date: 27/09/2023

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

Program

```
color_list1=[]
n=int(input("Enter number of colours in first list: "))
for i in range(0,n):
    e=input("Enter colours in first list : ")
    color_list1.append(e)
    color_list2=[]
n2=int(input("Enter number of colours in second list : "))
for i in range(0,n2):
    e2=input("Enter colours in second list : ")
    color_list2.append(e2)
    r = [color for color in color_list1 if color not in color_list2]
for color in r:
    print(color)
```

Output

```
Enter number of colours in first list: 4
Enter colours in first list : White
Enter colours in first list : Red
Enter colours in first list : Yellow
Enter colours in first list : Black
Enter number of colours in second list : 3
Enter colours in second list : Yellow
Enter colours in second list : Green
Enter colours in second list : Blue
White
Red
Black
```

COURSE OUTCOME 1

Date: 04/10/2023

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:
    list1 = list(string1)
    list2 = list(string2)
    list1[1], list2[1] = list2[1], list1[1]
    result = "".join(list1) + " " + "".join(list2)
else:
    result = string1 + " " + string2
print(result)
```

Output

```
Enter the first string : rachel
Enter the second string : rishon
richel rashon
```

COURSE OUTCOME 1

Date: 04/10/2023

17. Sort dictionary in ascending and descending order.

Program

```
d={'Rachel':3,'Rishon':4,'Jessy':2,'Jolly':1,'Anna':5}
print("Ascending order : ",dict(sorted(d.items())))
print("Descending order : ",dict(sorted(d.items(),reverse=True)))
```

Output

```
Ascending order :  {'Anna': 5, 'Jessy': 2, 'Jolly': 1, 'Rachel': 3, 'Rishon':
4}
Descending order :  {'Rishon': 4, 'Rachel': 3, 'Jolly': 1, 'Jessy': 2, 'Anna':
5}
```

COURSE OUTCOME 1

Date: 04/10/2023

18. Merge two dictionaries.

Program

```
d1={'Rachel':3,'Rishon':4,'Jessy':2,'Jolly':1}
d2={'Tina':9,'Nina':6,'Anna':5,'Tim':8,'Mike':7}
print("Merged dictionaries : ",d1|d2)
```

Output

```
Merged dictionaries :  {'Rachel': 3, 'Rishon': 4, 'Jessy': 2, 'Jolly': 1,
'Tina': 9, 'Nina': 6, 'Anna': 5, 'Tim': 8, 'Mike': 7}
```

COURSE OUTCOME 1

Date: 04/10/2023

19. Find gcd of 2 numbers.

Program

```
x=int(input("Enter the first number : "))
y=int(input("Enter the second number : "))
if x>y:
    x,y=y,x
for i in range(1,x+1):
    if x%i==0 and y%i==0:
        gcd=i
print("GCD of", x, "and", y, "is:", gcd)
```

Output

```
Enter the first number : 6
Enter the second number : 18
GCD of 6 and 18 is: 6
```

COURSE OUTCOME 1

Date: 04/10/2023

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

Program

```
n=int(input("Enter the number of integers : "))
l=[]
ol=[]
for i in range(1,n+1):
    num=int(input("Enter the number:"))
    l.append(num)
print("List before removal of even numbers : ",l)
ol=[i for i in l if i%2!=0]
print("List after removal of even numbers : ",ol)
```

Output

```
Enter the number of integers : 4
Enter the number:1
Enter the number:2
Enter the number:3
Enter the number:4
List before removal of even numbers :  [1, 2, 3, 4]
List after removal of even numbers :  [1, 3]
```

COURSE OUTCOME 2

Date: 09/10/2023

1. Program to find the factorial of a number

Program

```
n=int(input("Enter the number : "))
f=1;
if (n<0):
    print("Sorry, factorial does not exist")
elif (n==0):
    print("The factorial of 0 is 1")
else:
    for i in range(1,n+1):
        f = f*i
    print("The factorial of",n,"is",f)
```

Output

```
Enter the number : 5
The factorial of 5 is 120
```


COURSE OUTCOME 2

Date: 09/10/2023

2. Generate Fibonacci series of N terms.

Program

```
n=int(input("Enter the value for n : "))
x=0
y=1
sum=0
for i in range(0,n):
    x=y
    y=sum
    sum=x+y
    print(sum)
```

Output

```
Enter the value for n : 6
1
1
2
3
5
8
```

COURSE OUTCOME 2

Date: 09/10/2023

3. Find the sum of all items in a list.

Program

```
list = [2,6,8,9]
sum=sum(list)
print("Sum of all items in the list is ",sum)
```

Output

Sum of all items in the list is 25

COURSE OUTCOME 2

Date: 09/10/2023

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

```
r=[]
for i in range(1000,10000):
    if all(int(x)%2==0 for x in str(i)):
        if int(i**0.5)**2==i:
            r.append(i)
print("List of numbers : ",r)
```

Output

```
List of numbers :  [4624, 6084, 6400, 8464]
```

COURSE OUTCOME 2

Date: 09/10/2023

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

```
rows=int(input("Enter the number of rows : "))
i=1
while(i<=rows):
    j=1
    while(j<=i):
        print(i*j,end=' ')
        j=j+1
    print()
    i=i+1
```

Output

```
Enter the number of rows : 5
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

COURSE OUTCOME 2

Date: 11/10/2023

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

Program

```
s=input("Enter a string : ")
char_freq={}
print("Length of the string : ",len(s))
for char in s:
    if char.isalpha() or char.isspace():
        char=char.lower()
        char_freq[char]=char_freq.get(char,0)+1
for char,freq in char_freq.items():
    print(f"'{char}' occurs {freq} times")
```

Output

```
Enter a string : swimming
Length of the string :  8
's' occurs 1 times
'w' occurs 1 times
'i' occurs 2 times
'm' occurs 2 times
'n' occurs 1 times
'g' occurs 1 times
```

COURSE OUTCOME 2

Date: 11/10/2023

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

Program

```
s1=input("Enter a string : ")
if s1.endswith('ing'):
    s2=s1+'ly'
else:
    s2=s1+'ing'
print(s2)
```

Output

```
Enter a string : will
willing
Enter a string : willing
willingly
```

COURSE OUTCOME 2

Date: 11/10/2023

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

Program

```
n=int(input("Enter the size of the list : "))
print("Enter words in the list : ")
a=[input() for i in range(n)]
max=len(a[0])
temp=a[0]
for i in a:
    if len(i)>max:
        max=len(i)
        temp=i
print("The longest word is",temp,"with length ",max)
```

Output

```
Enter the size of the list : 5
Enter words in the list :
kitty
pin
inside
home
fit
The longest word is inside with length 6
```

COURSE OUTCOME 2

Date: 11/10/2023

9. Construct following pattern using nested loop

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

Program

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

Output

```
Enter a number : 5  
*  
**  
***  
****  
*****  
****  
***  
**  
*
```


COURSE OUTCOME 2

Date: 11/10/2023

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:56
The factors are:
1
2
4
7
8
14
28
56
```

COURSE OUTCOME 2

Date: 11/10/2023

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

Program

```
import math
s=int(input("Enter the side of the square : "))
area=lambda side:s*s
print("Area of the square : ",s*s)
a=int(input("Enter the length of the rectangle : "))
b=int(input("Enter the breadth of the rectangle : "))
area=lambda a,b:a*b
print("Area of the rectangle : ",a*b)
p=int(input("Enter the base of the triangle : "))
q=int(input("Enter the altitude : "))
area=lambda p,q:0.5*P*Q
print("Area of the triangle : ",0.5*p*q)
```

Output

```
Enter the side of the square : 4
Area of the square : 16
Enter the length of the rectangle : 2
Enter the breadth of the rectangle : 5
Area of the rectangle : 10
Enter the base of the triangle : 5
Enter the altitude : 6
Area of the triangle : 15.0
```

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

#graphics/rectangle.py

```
def area(length, width):  
    return length * width  
def perimeter(length, width):  
    return 2 * (length + width)
```

#graphics/circle.py

```
import math  
def area(radius):  
    return math.pi * radius**2  
def perimeter(radius):  
    return 2 * math.pi * radius
```

#graphics/threeDgraphics/cuboid.py

```
def surface_area(c_length, c_width, c_height):  
    return 2 * (c_length * c_width + c_width * c_height + c_height * c_length)  
def volume(c_length, c_width, c_height):  
    return c_length * c_width * c_height
```

#graphics/threeDgraphics/sphere.py

```
import math  
def surface_area(s_radius):  
    return 4 * math.pi * s_radius**2  
def volume(s_radius):  
    return (4/3) * math.pi * s_radius**3
```

COURSE OUTCOME 3

Date: 15/11/2023

#main.py

```
from graphics import rectangle, circle

from graphics.threeD_graphics import cuboid, sphere

print("Area and Perimeter of a Rectangle ")

length=int(input("Enter the length of the rectangle : "))

width=int(input("Enter the width of the rectangle : "))

print("Area of the Rectangle : ",rectangle.area(length, width))

print("Perimeter of the Rectangle : ",rectangle.perimeter(length, width))

print("Area and Perimeter of a Circle ")

radius=int(input("Enter the radius of the circle : "))

print("Area of the Circle : ",circle.area(radius))

print("Perimeter of the Circle : ",circle.perimeter(radius))

print("Area and Volume of a Cuboid ")

c_length=int(input("Enter the length of the cuboid : "))

c_width=int(input("Enter the width of the cuboid : "))

c_height=int(input("Enter the height of the cuboid : "))

print("Area of the Cuboid : ", cuboid.surface_area(c_length, c_width,

c_height))

print("Volume of the Cuboid : ", cuboid.volume(c_length, c_width, c_height))

print("Surface Area and Volume of a Sphere ")

s_radius=int(input("Enter the radius of the sphere : "))

print("Surface Area of the Sphere : ", sphere.surface_area(s_radius))

print("Volume of the Sphere : ", sphere.volume(s_radius))
```

COURSE OUTCOME 3

Date: 15/11/2023

Output

Area and Perimeter of a Rectangle

Enter the length of the rectangle : 3

Enter the width of the rectangle : 2

Area of the Rectangle : 6

Perimeter of the Rectangle : 10

Area and Perimeter of a Circle

Enter the radius of the circle : 4

Area of the Circle : 50.26548245743669

Perimeter of the Circle : 25.132741228718345

Area and Volume of a Cuboid

Enter the length of the cuboid : 5

Enter the width of the cuboid : 6

Enter the height of the cuboid : 4

Area of the Cuboid : 148

Volume of the Cuboid : 120

Surface Area and Volume of a Sphere

Enter the radius of the sphere : 5

Surface Area of the Sphere : 314.1592653589793

Volume of the Sphere : 523.5987755982989

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 compare_area(self, other_rectangle):
        if self.area() > other_rectangle.area():
            return "The first rectangle has a larger area."
        elif self.area() < other_rectangle.area():
            return "The second rectangle has a larger area."
        else:
            return "Both rectangles have the same area."
l1=int(input("Enter the length of rectangle1: "))
b1=int(input("Enter the breadth of rectangle1: "))
rectangle1 = Rectangle(l1, b1)
print("Area of rectangle1:",rectangle1.area())
print("Perimeter of rectangle1:",rectangle1.perimeter())
l2=int(input("Enter the length of rectangle2: "))
b2=int(input("Enter the breadth of rectangle2: "))
rectangle2 = Rectangle(l2, b2)
print("Area of rectangle2: ",rectangle2.area())
print("Perimeter of rectangle2: ",rectangle2.perimeter())
z =rectangle1.compare_area(rectangle2)
print(z)
```

Output

```
Enter the length of rectangle1: 3
Enter the breadth of rectangle1: 4
Area of rectangle1: 12
Perimeter of rectangle1: 14
Enter the length of rectangle2: 5
Enter the breadth of rectangle2: 3
Area of rectangle2: 15
Perimeter of rectangle2: 16
The second rectangle has a larger area.
```


COURSE OUTCOME 4

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, account_number, account_holder_name, account_type,
balance):
        self.account_number = account_number
        self.account_holder_name = account_holder_name
        self.account_type = account_type
        self.balance = balance
    def deposit(self, amount):
        if amount>0:
            self.balance += amount
            print("Deposit successful of ", amount)
            print("New balance = ",self.balance)
        else:
            print("Invalid deposit amount.")
    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)
ano=int(input("Enter account number : "))
name=input("Enter account holder : ")
atype=input("Enter account type : ")
amt=int(input("Enter account initial balance : "))
account1=BankAccount(ano,name,atype,amt)
account1.getbalance()
ch=0
while(ch!=4):
    print("\n\n1.Deposit amount\n2.Withdraw amount\n3.See account balance");
    ch=int(input("Enter 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 withdrawn : "))
        account1.withdraw(wamount)
        account1.getbalance()
    elif ch==3:
        account1.getbalance()
    else:
        print("Invalid");
```

COURSE OUTCOME 4

Date: 27/11/2023

Output

Enter account number : 3567421897
Enter account holder : Rishon Jolly
Enter account type : Savings
Enter account initial balance : 5000
Current balance = 5000

1. Deposit amount
2. Withdraw amount
3. See account balance
Enter choice : 1
Enter the amount to be deposited : 10000
Deposit successful of 10000
New balance = 15000

1. Deposit amount
2. Withdraw amount
3. See account balance
Enter choice : 2
Enter the amount to be withdrawn : 1000
Current balance = 14000

1. Deposit amount
2. Withdraw amount
3. See account balance
Enter choice : 3
Current balance = 14000

1. Deposit amount
2. Withdraw amount
3. See account balance
Enter choice : 4
Invalid

COURSE OUTCOME 4

Date: 27/11/2023

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, width):
        self._length = length
        self._width = width
    def area(self):
        return self._length * self._width
    def __lt__(self, other):
        return self.area() > other.area()

l1=int(input("Enter length of rectangle1 : "))
w1=int(input("Enter width of rectangle1 : "))
l2=int(input("Enter length of rectangle2 : "))
w2=int(input("Enter width of rectangle2 : "))
rectangle1 = Rectangle(l1,w1)
rectangle2 = Rectangle(l2,w2)
if rectangle1<rectangle2:
    print("Area of Rectangle 1 is smaller than the area of Rectangle 2.")
elif rectangle1>rectangle2:
    print("Area of Rectangle 1 is larger than the area of Rectangle 2.")
else:
    print("Both rectangles have the same area.")
```

Output

```
Enter length of rectangle1 : 3
Enter width of rectangle1 : 5
Enter length of rectangle2 : 2
Enter width of rectangle2 : 4
Area of Rectangle 1 is smaller than the area of Rectangle 2.
```

COURSE OUTCOME 4

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
        self._minute = minute
        self._second = second
    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}";
x1=int(input("Enter hour of first time:"))
y1=int(input("Enter minute of first time:"))
z1=int(input("Enter second of first time:"))
x2=int(input("Enter hour of next time:"))
y2=int(input("Enter minute of next time:"))
z2=int(input("Enter second of next time:"))
time1 = Time(x1,y1,z1)
time2 =Time(x2,y2,z2)
sum_time = time1 + time2
print("Time 1:", time1)
print("Time 2:", time2)
print("Sum of Time 1 and Time 2:",sum_time)
```

Output

```
Enter hour of first time:7
Enter minute of first time:05
Enter second of first time:55
Enter hour of next time:8
Enter minute of next time:2
Enter second of next time:4
Time 1: 07:05:55
Time 2: 08:02:04
Sum of Time 1 and Time 2: 15:07:59
```

COURSE OUTCOME 4

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):
        super().display_info()
        print("Price:", self.price)
        print("Number of Pages:", self.no_of_pages)
name=input("Enter Publisher:")
title=input("Enter title:")
author=input("Enter author:")
price=int(input("Enter price:"))
no_of_pages=int(input("Enter number of pages:"))
python_book = Python(name,title,author,price,no_of_pages)
python_book.display_info()
```

Output

```
Enter Publisher:Bloomsbury
Enter title:Harry Potter
Enter author:J K Rowling
Enter price:355
Enter number of pages:223
Publisher: Bloomsbury
Title: Harry Potter
Author: J K Rowling
Price: 355
Number of Pages: 223
```

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

Output

The contents of the file are

['Muthoot Institute of Technology and Science', 'Puthencruz, Varikoli,', 'Kochi, Kerala 682308',
'Website: mgmits.ac.in']

COURSE OUTCOME 5

Date: 16/10/2023

2. Python program to copy odd and even lines from one file to another

Program

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

Output

Original file data(stud.txt)

Muthoot Institute of Technology and Science
Puthencruz, Varikoli,
Kochi, Kerala 682308
Phone: 0484 273 2111
Website: mgmits.ac.in

Odd file data(odd.txt)

Muthoot Institute of Technology and Science
Kochi, Kerala 682308
Website: mgmits.ac.in

Even file data(even.txt)

Puthencruz, Varikoli,
Phone: 0484 273 2111

COURSE OUTCOME 5

Date: 16/10/2023

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("rna.csv","r") as efile:
    data=csv.reader(efile)
    for i in data:
        print(i)
```

Output

```
['Roll_no', 'Name', 'Age']
['1', 'Aby', '22']
['2', 'Ann', '21']
['3', 'Ren', '20']
['4', 'Ron', '21']
['5', 'Sen', '25']
```


COURSE OUTCOME 5

Date: 16/10/2023

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

Program

```
import csv
import pandas as pd
f = pd.read_csv("students.csv", usecols = ['Roll no', 'Name'])
print(f)
```

students.csv

Roll no	Name	Age
---------	------	-----

1	Ann	21
---	-----	----

2	Anu	22
---	-----	----

3	Bini	22
---	------	----

4	Riya	21
---	------	----

Output

Roll no	Name
---------	------

0	1 Ann
---	-------

1	2 Anu
---	-------

2	3 Bini
---	--------

3	4 Riya
---	--------

COURSE OUTCOME 5

Date: 16/10/2023

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=['Roll_no','Name','Age']
sdict=[{'Roll_no':'1','Name':'Ann','Age':'10'},
{'Roll_no':'2','Name':'Ani','Age':'10'},
{'Roll_no':'3','Name':'Don','Age':'11'},
{'Roll_no':'4','Name':'Jenny','Age':'11'}]
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

	Roll_no	Name	Age
0	1	Ann	10
1	2	Ani	10
2	3	Don	11
3	4	Jenny	11