COURSE OUTCOME1

DATE: 18/9/23

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

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 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 securityanalysis.

SonarQube (Community Edition) is an open source static + dynamic code analysis platform developed by Sonar Source for continuous inspection of codequality 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 year2023
enter final year2050
leap year from current year to final year is
2024
2028
2032
2036
2040
2044
2048
```

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

```
l1=[2,-4,5,6,-8]
l2=[i for i in l1 if i>0]
print("positive numbers:",l2)
```

OUTPUT

```
positive numbers: [2, 5, 6]
```

b) Square of N numbers

```
l1=[3,5,7]
l2=[i**2 for i in l1]
print("squares:",l2)
```

OUTPUT

```
squares: [9, 25, 49]
```

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

```
w=input("enter a word:")
v=[i for i in w if i in'aeiouAEIOU']
print("vowels:",v)
```

OUTPUT

```
enter a word: elephant

vowels: ['e', 'e', 'a']
```

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

```
c=input("enter a character: ")
o=[ord(i) for i in c]
print("ordinal value: ",o)
```

OUTPUT

```
enter a character: h
ordinal value: [104]
```

DATE: 18/09/2023

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

PROGRAM

```
n="this pen is an ink pen"
a=input("enter a word: ")
print(n.count('pen'))
```

OUTPUT

```
enter a word: pen
2
```

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

OUTPUT

```
enter the number of elements:3
enter the integer122
enter the integer44
enter the integer33
['over', 44, 33]
```

DATE:20/09/2023

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

PROGRAM

```
name=['meenu','pooja','binu']
for i in name:
print("count of element",i,":",i.count('a'))
```

OUTPUT

```
count of element meenu : 0
count of element pooja : 1
count of element binu : 0
```

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

```
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. [eg: python -> nythop]

PROGRAM

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

OUTPUT

```
nythop
```

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:"))
area=3.14*r*r
print("Area of circle is:",area)
```

OUTPUT

```
Enter the radius of the circle:2
Area of circle is: 12.56
```

DATE: 27/09/2023

11.Find biggest of 3 numbers entered

PROGRAM

```
a=int(input("enter first number:"))
b=int(input("enter second number:"))
c=int(input("enter third number:"))
if(a>b and a>c):
  print(a,"is large")
elif(b>a and b>c):
  print(b,"is large")
else:
  print(c,"is large")
```

OUTPUT

```
enter first number:2
enter second number:7
enter third number:0

7 is large
```

DATE: 27/09/2023

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

PROGRAM

```
F=input("enter a file:")
A=F.split(".")
print(A[1])
```

OUTPUT

```
enter a file:file.text

text
```

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=["violet" ,"red" ,"blue"]
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

```
The list of colours :  ['violet' , 'red', 'blue']
The first colour in the list is  violet

The last colour in the list is  blue
```

DATE: 27/09/2023

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

PROGRAM

```
n=int(input("Enter value for n : "))
a=n+(n*n)+(n*n*n)
print("Result : ",a)
```

OUTPUT

```
Enter value for n : 5
Result :  155
```

DATE: 27/09/2023

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

PROGRAM

```
c1=["Blue", "Green", "Yellow", "Red"]
c2=["Gray", "Yellow", "Pink", "Green", "Orange"]
res=[i for i in c2 if i not in c1]
print("Colors from color-list 1 not in color-list 2 : ",res)
```

OUTPUT

```
Colors from color-list 1 not in color-list 2 :
['Gray', 'Pink', 'Orange']
```

DATE: 04/10/2023

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

PROGRAM

```
s1=input("Enter string 1 : ")
s2=input("Enter string 2 : ")
s3=s2[0]+s1[1:]+" "+s1[0]+s2[1:]
print("Swapped string :",s3)
```

OUTPUT

```
Enter string 1 : meenu
Enter string 2 : pooja
Swapped string : peenu mooja
```

DATE: 04/10/2023

17. Sort dictionary in ascending and descending order.

PROGRAM

```
D= {'Meera': 10, 'Arun': 20,'Manu': 30, 'Sanny': 40,
'Sonu': 50}
print("Ascending order : ",dict(sorted(d.items())))
print("Descending order : ",dict(sorted(d.items(),
reverse=True)))
```

OUTPUT

```
Ascending order :  {'Arun': 20, 'Manu': 30, 'Meera':
10, 'Sanny': 40, 'Sonu': 50}

Descending order :  {'Sonu': 50, 'Sanny': 40, 'Meera':
10, 'Manu': 30, 'Arun': 20}
```

DATE: 04/10/2023

18. Merge two dictionaries.

PROGRAM

```
D1 = {'Arohi': 10,'Vishnu': 20, 'Abhi': 30}
D2 = {'Saji': 20,'Divya': 40, 'Leena': 50}
print("Merged dictionaries :",D1|D2)
```

OUTPUT

```
Merged dictionaries : {'Arohi': 10, 'Vishnu': 20,
'Abhi': 30, 'Saji': 20, 'Divya': 40, 'Leena': 50}
```

DATE: 04/10/2023

19. Find gcd of 2 numbers.

PROGRAM

```
a=int(input("enter first number:"))
b=int(input("enter second number:"))
if(a<b):
  aib=a
else:
  aib=b
for i in range(1,aib+1):
  if((a%i==0)and(b%i==0)):
    gcd=i
print("gcd",gcd)
```

OUTPUT

```
enter first number:752
enter second number:1000
gcd 8
```

DATE: 04/10/2023

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

PROGRAM

```
n=int(input("enter n:"))
li=[]
lo=[]
for i in range(1,n+1):
 num=int(input("enter  number:"))
 li.append(num)
print("list before removing:",li)
lo=[i for i in li if i%2!=0]
print("list after removing:",lo)
```

OUTPUT

```
enter n:3
enter  number:5
enter  number:9
enter  number:6
list before removing: [5, 9, 6]
list after removing: [5, 9]
```

COURSE OUTCOME-2

DATE:09/10/2023

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=",fact)
```

OUTPUT

```
enter a number:4
factorial= 24
```

DATE: 09/10/2023

2.Generate Fibonacci series of N terms

PROGRAM

```
t1=0
t2=1
n=int(input("enter a limit:"))
print("fibanacci 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
fibanacci series is:
0
1
1
2

3
```

DATE: 09/10/2023

3. Find the sum of all items in a list

PROGRAM

```
li=[3,6,7,8,99]
sum=0
for i in li:
 sum=i+sum
print("sum of li=",sum)
```

OUTPUT

```
sum of li= 123
```

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

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

OUTPUT

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

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 total 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 total Number of Rows : 4
1
2 4
3 6 9
4 8 12 16
```

DATE:11/10/2023

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

PROGRAM

```
text ="Good evening"
char_freq = {}
print("Total length of string: ",len(text));
for char in text:
 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

```
Total length of string:  12
'g' occurs 2 times
'o' occurs 2 times
'd' occurs 1 times
' ' occurs 1 times
'e' occurs 2 times
'v' occurs 1 times
'n' occurs 2 times
'i' occurs 1 times
```

DATE: 11/10/2023

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

PROGRAM

```
str1=input("Enter a string : ")
if str1.endswith('ing'):
 str2=str1+'ly'
else:
 str2=str1+'ing'
print(str2)
```

OUTPUT

```
Enter a string : reading
Readingly
Enter a string : read
reading
```

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 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("Word with max length is",temp,"It's length is",max)
```

OUTPUT

```
Enter the size of the list : 3
Enter list :
sunday
dancing
tutorial
Word with max length is tutorial It's length is 8
```

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

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

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:8
The factors are:
1
2
4

8
```

DATE: 11/10/2023

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

PROGRAM

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

OUTPUT

```
enter side:2
area of square: 4
enter length:4
enter breadth:3
area of rectangle: 12
enter base:4
enter altitude:6
area of triangle: 12.0
```