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 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 final year entered by user.

Program:

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

print("list of leap year between 2023 and " + str(endyr)+":")

for i in range(2023,endyr):

if (i % 4 ==0 and i % 100 !=0) or (i % 400 ==0):

        print(i)

        i+=1
```

Output:

```
enter end year:2050

list of leap year between 2023 and 2050:

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.

Program:

```
list1 = [11, -21, 0, 45, 66, -93]

list2 = [i for i in list1 if i>0]

print("positive list of numbers",list2)
```

Output:

```
positive list of numbers [11, 45, 66]
```

(b) Square of N numbers.

Program:

```
l2 = [1,2,5,-3,9,10]

l3 = [i**2 for i in l2]

print("square of numbers",l3)
```

Output:

```
square of numbers [1, 4, 25, 9, 81, 10
```

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

Program:

```
s=input("enter a word:")

l=[i for i in s if i in 'aeiouAEIOU']

print(l)
```

Output:

```
enter a word:animal

['a', 'i', 'a']
```

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

Program:

```
print("hello world")

s=input("enter a character:")

m=[ord(i) for i in s]

print(m)
```

Output:

```
hello world

enter a character:rdx

[114, 100, 120]
```

COURSE OUTCOME 1

Date: 18/09/2023

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

Program:

```
str=input("Enter a sentence : ")

a=[]

a=str.split();

words=[a.count(i) for i in a]

print(dict(zip(a,words)))
```

Output:

```
Enter a sentence : today is monday next monday is my birthday

{'today': 1, 'is': 2, 'monday': 2, 'next': 1, 'my': 1, 'birthday': 1}
```

COURSE OUTCOME 1

Date: 20/09/2023

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(a)
 else:
   list.append("over")
print(list)
```

Output:

```
enter the number of elements:4
enter the integer:34
enter the integer:105
enter the integer:23
enter the integer:110
[34, 'over', 23, 'over']
```

COURSE OUTCOME 1

Date: 20/09/2023

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

Program:

```
names=['sahala','suhana','beema','karun']

for i in names:

 print("'a' occurs in ", i,i.count('a'),'times')
```

Output:

```
'a' occurs in  sahala 3 times

'a' occurs in  suhana 2 times

'a' occurs in  beema 1 times

'a' occurs in  karun 1 times
```

COURSE OUTCOME 1

Date: 20/09/2023

7.Enter 2 lists of integers. Check :

(a) Whether list are of same length.

Program:

```
list1=[3,5,2,1,6]

list2=[6,3,2,1,5]

if len(list1) == len(list2):

 print("lists are of same length")

else:

 print("they are different")
```

Output:

```
lists are of same length
```

(b) whether list sums to same value

Program:

```
list1=[3,5,2,1,6]

list2=[6,3,2,1,5]

if sum(list1) == sum(list2):

 print("sum has same value")

else:

 print("sum has different value")
```

Output:

```
sum has same value
```

(c) whether any value occur in both

Program:

```
list1=[4,9,0,1,8]

list2=[6,3,2,1,5]

output = any(check in list1 for check in list2)

if output:

  print(x,y, 'present in lists')

else :

  print("not prsent")
```

Output:

```
1 present in list
```

COURSE OUTCOME 1

Date: 20/09/2023

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

Program:

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

Output:

```
Enter a string: onion
Replaced string: oni$n
```

COURSE OUTCOME 1

Date: 27/09/2023

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

Program:

```
s = input('Enter a string: ')

s[-1]+s[1:-1]+s[0]
```

Output:

```
Enter a string: sahala

‘aahals’
```

COURSE OUTCOME 1

Date: 27/09/2023

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

Program:

```
r=int(input("Enter the radius:"))

Area=3.14*r*r

print("area of circle is:",Area)
```

Output:

```
Enter the radius:4

area of circle is: 50.24
```

COURSE OUTCOME 1

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 greater")
elif b>a and b>c:
 print(b,"is greater")
else:
 print(c,"is greater")
```

Output:

```
Enter first number:5
Enter second number:6
Enter third number:2
6 is greater
```

COURSE OUTCOME 1

Date: 27/09/2023

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

Program:

```
f=input("Enter a filename: ")

e=f.split(".")

print(e[1])
```

Output:

```
Enter a filename: sahala.py

py
```

COURSE OUTCOME 1

Date: 04/10/2023

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 in the list: "))
list=[]
print("Enter the color: ")
for i in range(n):
 c=input()
 list.append(c)
 print(list)
print(list[0],list[-1])
```

Output:

```
Enter the number of colors in the list: 4
Enter the color:
orange
['orange']
blue
['orange', 'blue']
red
['orange', 'blue', 'red']
black
['orange', 'blue', 'red', 'black']
orange black
```

COURSE OUTCOME 1

Date: 04/10/2023

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

Program:

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

s = n+(n*11)+(n*111)

print(str(n)+ "+"+str(n*11)+"+"+str(n*111)+"+"+"is"+str(s))
```

Output:

```
Enter a number: 3 3+33+333+is 369
```

COURSE OUTCOME 1

Date: 04/10/2023

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

Program:

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

Output:

```
Colors from color-list 1 not in color-list 2: ['Purple', 'Pink', '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:

```
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 : mahala
Enter string 2 : sol
Swapped string : sahala mol
```

COURSE OUTCOME 1

Date: 09/10/2023

17. Sort dictionary in ascending and descending order.

Program:

```
d = {'Allen': 1, 'Eden': 15, 'David': 12, 'Kiran': 33}

ascending_order = dict(sorted(d.items()))

descending_order = dict(sorted(d.items(), reverse=True))

print("Ascending order:", ascending_order)

print("Descending order:", descending_order)
```

Output:

```
Ascending order: {'Allen': 1, 'David': 12, 'Eden': 15, 'Kiran': 33}
Descending order: {'Kiran': 33, 'Eden': 15, 'David': 12, 'Allen': 1}
```

COURSE OUTCOME 1

Date: 09/10/2023

18. Merge two dictionaries.

Program:

<u>Method-1 :</u>

```
d1={"name":"mamma","age":24, "gender":"male"}
d2= {"height":170}
print(d1|d2)
```

Output:

```
{'name': 'mamma', 'age': 24, 'gender': 'male', 'height': 170}
```

<u>Method-2 :</u>

```
d1={"name":"sahala","age":21, "gender":"female"}

d2= {"height":170}

m={**d1,**d2}

print(m)
```

Output:

```
{'name': 'sahala', 'age': 21, 'gender': 'female', 'height': 170}
```

COURSE OUTCOME 1

Date: 09/10/2023

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)):

    if num1 % i == 0 and num2 % i == 0:

        gcd = i

print("GCD of", num1, "and", num2, "is", gcd)
```

Output:

```
Enter the first number: 42
Enter the second number: 22
GCD of 42 and 22 is 2
```

COURSE OUTCOME 1

Date: 09/10/2023

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

Program:

```
list=[10,1,8,21,89,32,23,59]

print("The old list: ")

print(list)

for i in list:

  if i % 2==0:

    list.remove(i)

print("List after removing even numbers:")

print(list)
```

Output:

```
The old list: [10, 1, 8, 21, 89, 32, 23, 59]
 List after removing even numbers: [1, 21, 89, 23, 59]
```

COURSE OUTCOME 2

Date: 11/10/2023

1. Program to find the factorial of a number

Program:

```
n=int(input("Enter a number:"))
fact=1
for i in range(1,n+1):
  fact=fact*i
print("factorial of ",n,"is",fact)
```

Output:

```
Enter a number:6
factorial of 6 is 720
```

COURSE OUTCOME 2

Date: 11/10/2023

2. Generate Fibonacci series of N terms

Program:

```
a=0
b=1
c=0
n=int(input("Enter a limit:"))
print("Fibonacci series are:")
for i in range(0,n):
  print(c,end ="")
  a=b
  b=c
  c=a+b
```

Output:

```
Enter a limit:10

Fibonacci series are: 0 1 1 2 3 5 8 1 3 2 1 3 4
```

COURSE OUTCOME 2

Date: 11/10/2023

3. Find the sum of all items in a list

Program:

```
list=[]
s=0
n=int(input("Enter size of list: "))
for i in range(0,n):
 a=int(input())
 list.append(a)
 s=s+a
print("Sum of all items in list:",s)
```

Output:

```
Enter size of list: 4
3
5
7
6
Sum of all items in list: 21
```

COURSE OUTCOME 2

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:

```
import math
u=int(input("Enter upper range: "))
for i in range(1000,u+1):
    if i%2==0:
      sqrt_num = math. sqrt(i)
      if sqrt_num. is_integer():
              print(i)
```

Output:

```
Enter upper range: 2000
 1024
 1156
 1296
 1444
 1600
 1764
 1936
```

COURSE OUTCOME 2

Date: 16/10/2023

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

Output:

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

COURSE OUTCOME 2

Date: 16/10/2023

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

Program:

```
n=input("Enter the String: ")
s={}
for i in n:
 if i in s:
  s[i]+=1
 else:
    s[i]=1
print(s)
```

Output:

```
Enter the String: sahala
{'s': 1, 'a': 3, 'h': 1, 'l': 1}
```

COURSE OUTCOME 2

Date: 16/10/2023

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

Program:

```
s=input("Enter the String: ")
if s[-3:] == 'ing':
 print(s + 'ly')
else:
 print(s + 'ing')
```

Output:

```
Enter the String: drawing
drawingly
```

```
Enter the String: play
playing
```

COURSE OUTCOME 2

Date: 16/10/2023

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

Program:

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

Output:

```
Enter size of list: 4
sahala is good girl
The word with the longest length is: sahala
```

COURSE OUTCOME 2

Date: 18/10/2023

9. Construct following pattern using nested loop

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

Program:

```
n = int(input("Enter number of rows: "))

for i in range(n):

 for j in range(i):

  print ('* ', end="")

 print('')

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

 for j in range(i):

  print('* ', end="")

 print('')
```

Output:

```
Enter number of rows: 5

*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

COURSE OUTCOME 2

Date: 18/10/2023

10. Generate all factors of a number.

Program:

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

Output:

```
Enter a number: 100
1
2
4
5
10
20
25
50
100
```

COURSE OUTCOME 2

Date: 18/10/2023

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

Program:

```
len=int(input("enter the length of the rectangle:"))

bth=int(input("enter the breadth of the rectangle:"))

rect_area = lambda len, bth : len*bth

a=int(input("enter the side of the square:"))

square_area = lambda a : a*a

base=int(input("enter the base of the triangle:"))

height=int(input("enter the height of the triangle:"))

tri_area = lambda base, height : 0.5*base*height

print("area of rectangle=",rect_area(len,bth))

print("area of square=",square_area(a))

print("area of triangle=",tri_area(base,height))
```

Output:

```
enter the length of the rectangle:4

enter the breadth of the rectangle:5

enter the side of the square:4

enter the base of the triangle:2

enter the height of the triangle:4

area of rectangle= 20

area of square= 16

area of triangle= 4.0
```