COURSE OUTCOME 1

DATE:1/10/24

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

- Customer 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

:

DATE 8/10/24

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

PROGRAM

```
y1=int(input("Enter a year:"))
y2=int(input("Enter final year:"))
for i in range(y1,y2+1):
if i%4==0 and i%100!=0 or i%400==0:
print(i)
```

OUTPUT

```
Enter a year:2024
Enter final year:2036
2024
2028
2032
2036

Enter a year:2003
Enter final year:2011
2004
2008
```

DATE:10/10/24

3..List comprehensions:

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

PROGRAM

```
l=input("Enter list of integer seperated by spaces:")
l1=[int(n) for n in l.split()]

print(l1)

p=[n for n in l1 if n>0]
print("The list of positive nos. are",p)
```

OUTPUT

```
Enter list of integer seperated by spaces:4 8 3 -5 -6 -7 10
[4, 8, 3, -5, -6, -7, 10]
The list of positive nos. are [4, 8 3, 10]
```

b.Square of N numbers

PROGRAM

```
l2=input("Enter the list of numbers:")
l3=[int(n) for n in l2.split()]
print(l3)
```

```
s=[n*n for n in l3 ]
print("Square of n mumber",s)
```

OUTPUT

```
Enter the list of numbers: 2 3 4 6
[ 2, 3, 4, 6]
Square of n mumber [4, 9, 16, 36]
```

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

PROGRAM

```
word=input("Enter a Word:")
w1=[n for n in word]
print(w1)
v=['a','e','i','o','u']
l=[n for n in w1 if n in v]
print("Vowels:",l)
```

OUTPUT

```
Enter a Word:hello
['h', 'e', 'l', 'l', 'o']
Vowels: ['e', 'o']
```

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

PROGRAM

```
list=input("Enter a word:")
a=[n for n in list]
print(a)
ord=[ord(n) for n in a]
print(" List ordinal value of each element of a word:\n",ord);
```

OUTPUT

```
Enter a word:hello
['h', 'e', 'l', 'l', 'o']
List ordinal value of each element of a word:
[104, 101, 108, 108, 111]
```

DATE:7/11/24

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

PROGRAM

```
text=[word for word in input("Enter a text:").split()]
count=[(word, text.count(word)) for word in set(text)]
print("Word occurrences:")
for word, cnt in count:
print(word + ":", cnt)
```

OUTPUT

```
Enter a text:welcome to python
Word occurrences:
to: 1
python: 1
welcome: 1

Enter a text:hai hello hey
Word occurrences:
hai: 1
hey: 1
hello: 1
```

DATE:15/10/24

5.Prompt the user for a list of integers . for all values greter than 100 store 'over' instead use list comprehension

PROGRAM

```
u=input("Enter the integers:")
result=['over' if int(num) > 100 else int(num) for num in u.split()]
print(result)
```

OUTPUT

```
Enter the integers:5 77 999 1000
[5, 77, 'over', 'over']

Enter the integers:999 555 44 99 100
['over', 'over', 44, 99, 100]
```

DATE:10/10/24

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

PROGRAM

```
names=input("Enter a list of first names separated by spaces: ").split()
c=0
for name in names:
c+= name.lower().count('a')
print("The total occurrences of 'a':", c)
```

OUTPUT

```
Enter a list of first names separated by spaces: tessa jake roselin jimmy
The total occurrences of 'a': 2

Enter a list of first names separated by spaces: anu manu sanu linu
The total occurrences of 'a': 3
```

DATE:7/11/24

7.Enter 2 lists of integers. Check

PROGRAM

a. Whether list are of same length

```
list1=list(input("Enter values in list 1: ").split())
list2=list(input("Enter values in list 2: ").split())
print("Length of list 1:",len(list1))
print("Length of list 2:",len(list2))
if(len(list1)==len(list2)):
print("The list are same length of",len(list1))
else:
print("The list are not equal length")
```

OUTPUT

```
Enter values in list 2: 9 1 8 2 7
Length of list 1: 5
Length of list 2: 5
The list are same length of 5

Enter values in list 2: 9 6 4
Length of list 1: 4
Length of list 2: 3
The list are not equal length
```

b. whether list sums to same value

PROGRAM

```
l1= [int(n) for n in input("Enter values in list 1: ").split()]
l2= [int(n) for n in input("Enter values in list 2: ").split()]
s1=sum(l1)
s2=sum(l2)
print("Sum of list1:",s1)
print("Sum of list2:",s2)
if(s1==s2):
print("The sum of list1 & list2 are same value:",s1)
else:
print("The sum are not equal")
```

OUTPUT

```
Enter values in list 1: 5 4 3 2 1
Enter values in list 2: 1 2 3 4 5
Sum of list1: 15
Sum of list2: 15
The sum of list1 & list2 are same value: 15

Enter values in list 2: 9 5 7 3 4
Sum of list1: 26
Sum of list2: 28
The sum are not equal
```

c. whether any value occur in both

PROGRAM

```
l1=list(input("Enter values in list 1: ").split())
l2=list(input("Enter values in list 2: ").split())
value=set(l1)&set(l2)
if value:
print("Common values between the lists:",value)
else:
print("No common values")
```

OUTPUT

```
Enter values in list 1: 5 7 9 3 9
Enter values in list 2: 9 4 6 3 1
Common values between the lists: {'9', '3'}

Enter values in list 1: 10 20 30 40 50
Enter values in list 2: 11 21 31 41 51
No common values
```

DATE:24/10/24

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

PROGRAM

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

OUTPUT

```
Enter a string:hai hello
hai $ell

Enter a string:Perfect Python projects promote progress
perfect $ython $rojects $romote $rogres
```

DATE:24/10/24

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

PROGRAM

```
s=input("Enter a string:")
s2=s[-1]+s[1:-1]+s[0]
print("String after first and last characters exchanged:",s2)
```

OUTPUT

```
Enter a string:hello
String after first and last characters exchanged: oellh

Enter a string:python
String after first and last characters exchanged: nythop
```

DATE:1/10/24

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

PROGRAM

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

OUTPUT

```
Enter the radius value:6
Area of circle= 113.03999999999999

Enter the radius value:77
Area of circle= 18617.06
```

DATE: 1/10/24

11.Find biggest of 3 numbers entered.

PROGRAM

```
x=int(input("Enter x value:"))
y=int(input("Enter y value:"))
r=int(input("Enter r value:"))
if y>x and y>r:
print("y is greatest")
elif x>y and x>r:
print("x is greatest")
else :
print(" r is greatest")
```

OUTPUT

```
Enter x value:77
Enter y value:99
Enter r value:44
y is greatest

Enter x value:44
Enter y value:11
Enter r value:88
r is greatest
```

DATE:24/10/24

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

PROGRAM

```
filename=input("Enter a file name:")
ext=filename.split('.')[-1]
print("The extension of the file is:", ext)
```

OUTPUT

```
Enter a file name:photo.png
The extension of the file is: png

Enter a file name:python.py
The extension of the file is: py
```

DATE:15/10/24

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

PROGRAM

```
l1=input("Enter list of colors with comma-separated color:").split(",")
print("Display first color: ",l1[0])
print("Display last color: ",l1[-1])
```

OUTPUT

```
Enter list of colors with comma-separated color:pink,blue,red,black
Display first color:  pink
Display last color:  black

Enter list of colors with comma-separated color:Blue,Violet,Yellow,Indigo
Display first color:  Blue
Display last color:  Indigo
```

DATE:3/10/24

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

PROGRAM

```
n = int(input("Enter value of n: "))
nn = int(str(n)*2)
nnn = int(str(n)*3)
result = n + nn + nnn
print("n + nn + nnn =",n,"+",nn,"+",nnn,"=", result)
```

OUTPUT

```
Enter value of n: 4
n + nn + nnn = 4 + 44 + 444 = 492

Enter value of n:5
n+nn+nnn = 5+55+555=615
```

DATE:15/10/24

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

PROGRAM

```
l1 = input("Enter colors for List 1, separated by commas: ").split(",")
l2 = input("Enter colors for List 2, separated by commas: ").split(",")
print("List 1:", l1)
print("List 2:", l2)
result = set(l1) - set(l2)
print("Print Colors in List 1 but not in List 2 as Set:", result)
print("Print as List:", list(result))
```

OUTPUT

```
List 1: ['red', 'yellow', 'blue', 'black']
List 2: ['red', 'purple', 'green']
Print Colors in List 1 but not in List 2 as Set: {'yellow', 'blue', 'black'}
Print as List: ['yellow', 'blue', 'black']

List 1: ['red', 'black', 'blue']
List 2: ['green', 'black', 'blue']
Print Colors in List 1 but not in List 2 as Set: {'red'}
Print as List: ['red']
```

DATE:15/10/24

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

PROGRAM

```
s1=input("Enter string1:")
s2=input("Enter string2:")
s3=s2[0]+s1[1:]+" "+s1[0]+s2[1:]
print("Modified string:",s3)
```

OUTPUT

```
Enter string1:hello
Enter string2:world
Modified string: wello horld

Enter string1:my name
Enter string2:is tessa
Modified string: iy name  ms tessa
```

DATE:22/10/24

17.Sort dictionary in ascending and descending order.

PROGRAM

```
d={"apple":10,"kiwi":20,"grape":12,"banana":23}
print("Dictionary before sorting:",d)
aresult=dict(sorted(d.items()))
print("Dictionary in ascending order:",aresult)
bresult=dict(sorted(d.items(),reverse=True))
print("Dictionary in descending order:",bresult)
```

OUTPUT

```
Dictionary before sorting: {'apple': 10, 'kiwi': 20, 'grape': 12, 'banana': 23}
Dictionary in ascending order: {'apple': 10, 'banana': 23, 'grape': 12, 'kiwi': 20}
Dictionary in descending order: {'kiwi': 20, 'grape': 12, 'banana': 23, 'apple': 10}
```

DATE:22/10/24

18.Merge two dictionaries.

PROGRAM

```
d1 = {}
d2 = {}
for i in range(int(input("Enter number of items for d1: "))):
key = input("Enter key: ")
value = int(input("Enter value: "))
d1[key] = value

for i in range(int(input("Enter number of items for d2: "))):
key = input("Enter key: ")
value = int(input("Enter value: "))
d2[key] = value

print("d1:", d1)
print("d2:", d2)
d1.update(d2)
print("After Update: ",d1)
print("Merging using pipe symbol'|': ",d1|d2)
print("Merging using '**':",{**d1,**d2})
```

OUTPUT

```
Enter number of items for d1: 2
Enter key: apple
Enter value: 4
Enter key: kiwi
Enter value: 7
Enter number of items for d2: 4
Enter key: orange
Enter value: 8
Enter key: cherry
Enter value: 1
Enter key: berry
Enter value: 7
Enter key: grapes
Enter value: 8
d1: {'apple': 4, 'kiwi': 7}
d2: {'orange': 8, 'cherry': 1, 'berry': 7, 'grapes': 8}
After Update:  {'apple': 4, 'kiwi': 7, 'orange': 8, 'cherry': 1, 'berry': 7, 'grapes': 8}
Merging using pipe symbol'|':  {'apple': 4, 'kiwi': 7, 'orange': 8, 'cherry': 1,
'berry': 7, 'grapes': 8}
Merging using '**': {'apple': 4, 'kiwi': 7, 'orange': 8, 'cherry': 1, 'berry': 7,
'grapes': 8}
```

DATE:29/10/24

19.Find gcd of 2 numbers.

```
import math
x=int(input("Enter first number:"))
y=int(input("Enter second number:"))
print("GCD of two number=",math.gcd(x,y))
```

OUTPUT

```
Enter first number:7
Enter second number:9
GCD of two number= 1

Enter first number:20
Enter second number:18
GCD of two number= 2
```

DATE:7/11/24

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

PROGRAM:

```
list=[int(i) for i in input("Enter the integers :").split()]
nl=[i for i in list if i%2!=0]
print("List of integers after removing even nos.:",nl)
```

OUTPUT

```
Enter the integers :7 9 4 6 1 3
List of integers after removing even nos.: [7, 9, 1, 3]

Enter the integers :88 71 92 11 31
List of integers after removing even nos.: [71, 11, 31]
```

COURSE OUTCOME 2

DATE: 3/10/24

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*=i
print("Factorial of",n,":",fact)
```

OUTPUT

```
Enter a number:7
Factorial of 7 : 5040

Enter a number:5
Factorial of 5:120
```

DATE:3/10/24

2.Generate Fibonacci series of N terms

PROGRAM

```
N=int(input("Enter the limit:"))
a=0
b=1
print(a)
print(b)
for i in range(2,N):
sum=a+b
print(sum)
a=b
b=sum
```

OUTPUT

```
Enter the limit:7
0
1
1
2
3
5
```

```
8
```

```
Enter the limit:4
0
1
1
2
```

DATE:8/10/24

3.Find the sum of all items in a list

PROGRAM

```
list = input("Enter the list elements separated by space: ").split()
list = [int(n) for n in list]
print("List =", list)
sum = 0
for i in range(len(list)):
sum += list[i]
print("Sum of items in the list:", sum)
```

OUTPUT

```
Enter the list elements separated by space: 6 9 4 6 3
List = [6, 9, 4, 6, 3]
Sum of items in the list: 28

Enter the list elements separated by space: 1 2 3 4 5 6
List = [1, 2, 3, 4, 5, 6]
Sum of items in the list: 21
```

DATE:24/10/24

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

```
f = int(input("Enter the starting number: "))
l = int(input("Enter the ending number: "))
result = []
for i in range(f, l + 1):
if int(i ** 0.5) ** 2 == i:
even = 0
for n in str(i):
if int(n) % 2 != 0:
even = 1
break
if even == 0:
result.append(i)
print("Numbers that are perfect squares and have all even digits:", result)
```

OUTPUT

```
Enter the starting number: 100
Enter the ending number: 800
Numbers that are perfect squares and have all even digits: []

Enter the starting number: 7000
Enter the ending number: 9000
Numbers that are perfect squares and have all even digits: [8464]
```

DATE:8/10/24

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 a limit:"))
for i in range(1,N+1):
for j in range(1,i+1):
print(i*j,end=" ")
print()
```

OUTPUT

```
Enter a limit:8
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49
8 16 24 32 40 48 56 64
```

```
Enter a limit:4
1
2 4
3 6 9
4 8 12 16
```

DATE:24/10/24

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

PROGRAM

```
s = input("Enter a string: ")
count = {}
for ch in s:
if ch in count:
count[ch] += 1
else:
count[ch] = 1
for ch, cnt in count.items():
print(ch + ":", cnt)
```

OUTPUT

```
Enter a string: hello
h: 1
e: 1
l: 2
o: 1
Enter a string: welcome
w: 1
e: 2
l: 1
c: 1
o: 1
m: 1
```

DATE:24/10/24

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

PROGRAM

```
s = input("Enter a string: ")
if len(s) >= 3:
if s[-3:] == "ing":
s += "ly"
else:
s += "ing"
print("Modified string:",s)
```

OUTPUT

```
Enter a string: programing
Modified string: programingly

Enter a string: python
Modified string: pythoning
```

DATE:29/10/24

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

PROGRAM

```
list=[words for words in input("Enter the list words:").split()]
longest=list[0]
for word in list:
if (len(word)> len(longest) ):
longest=word
print("Longest word in list",longest)
print("Length of longest word",len(longest))
```

OUTPUT

```
Enter the list words:apple banana berry pomogranate
Longest word in list pomogranate
Length of longest word 11

Enter the list words:table book chair tablesheet
Longest word in list tablesheet
Length of longest word 10
```

DATE:1/10/24

9.Construct following pattern using nested loop

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

PROGRAM

```
n=int(input("Enter a limit:"))
for i in range(1,n+1):
print("* "*i)
for j in range(n-1,0,-1):
print("* "*j)
```

OUTPUT

```
Enter a limit:6
*
* *
* * *
* * * *
* * * * *
* * * * * *

* * * * *

* * * *

* * *

* *

*

Enter a limit:3
*

* *

* * *

* *

*
```

DATE:29/10/24

10.Generate all factors of a number

PROGRAM

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

OUTPUT

```
Enter a number: 7
Factors of 7 :
1
7

Enter a number: 20
Factors of 20 :
1
2
4
5
10
20
```

DATE:29/10/24

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

PROGRAM

```
area1=lambda a : a*a
area2=lambda l,b : l*b
area3=lambda b,h: 0.5*b*h
s=int(input("Enter the side of the Square:"))
print("Area of Square",area1(s))
l=int(input("Enter the length of Rectangle:"))
b=int(input("Enter the breadth of Rectangle:"))
print("Area of Rectangle",area2(l,b))
h=int(input("Enter the height of Triangle:"))
b=int(input("Enter the base of Triangle:"))
print("Area of Triangle",area3(l,b))
```

OUTPUT

```
Enter the side of the Square:6
Area of Square 36

Enter the length of Rectangle:12
Enter the breadth of Rectangle:10
Area of Rectangle 120

Enter the height of Triangle:18
Enter the base of Triangle:14
Area of Triangle 84.0




Enter the side of the Square:4
Area of Square 16

Enter the length of Rectangle:8
Enter the breadth of Rectangle:5
Area of Rectangle 40

Enter the height of Triangle:5
Enter the base of Triangle:2
Area of Triangle 8.0
```