COURSE OUTCOME 1

DATE: 26/09/2024

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.

DATE : 03/10/2024

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

PROGRAM

```
cy = int(input("Enter Current Year : "))
ey = int(input("Enter Ending Year : "))
print("Leap years are : ")
for  i in range(cy,ey):
        if i % 400 == 0 and i % 100 == 0:
                print(i)
        elif i % 4 == 0 and i % 100 != 0:
                print(i, end = " ")
```

OUTPUT

```
Enter Current Year : 2020
Enter Ending Year : 2040
Leap years are : 2020  2024  2028 2032 2036 2040
```

```
Enter Current Year : 2024
Enter Ending Year : 2050
Leap years are : 2024  2028  2032  2036  2040  2044  2048
```

DATE : 03/10/2024

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**

PROGRAM (a)

```
l=[int(i) for i in input("Enter list of integers : ").split()]
p=[i for i in l if i>=0]
print("Positive Integers : ",p)
```

OUTPUT

```
Enter list of integers : 2 4 3 -5 –9 8
Positive Integers : [2, 4, 3, 8]

Enter List of integers : -6 -3 6 7 –2 4
Positive Integers : [6, 7, 4]
```

PROGRAM (b)

```
l=[int(i) for i in input("Enter List : ").split()]
l1=[i*i for i in l]
print(l1)
```

OUTPUT

```
Enter List : 1 2 3 4 5 6 7
[1, 4, 9, 16, 25, 36, 49]

Enter List : 5 6 7 8 9
[25, 36, 49, 64, 81]
```

PROGRAM (c)

```
word=input("Enter word : ")
vowels = "aeiouAEIOU"
vowel_list = [i for i in word if i in vowels]
print("Vowels in ",word," : ",vowel_list)
```

OUTPUT

Enter word : Programming
Vowels in Programming : [‘o’, ‘a’, ‘i’]

Enter word : Hello
Vowels in Hello : [‘e’, ‘o’]

PROGRAM (d)

```
word=input("Enter a word : ")
ordinal_values = [ord(char) for char in word]
print(f"The ordinal values of the characters are : {ordinal_values}")
```

OUTPUT

Enter a word : Python
The ordinal values of the characters are : [80, 121, 116, 104, 111, 110]

Enter word : Hello
The ordinal values of the characters are : [72, 101, 108, 108, 111]

DATE : 08/10/2024

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

PROGRAM

```
l=input("Enter a line : ")
words = l.split()
res = {}
for word in words:
    word = word.lower()
    if word in res:
        res[word] += 1
    else:
        res[word] = 1

for word, count in res.items():
    print(f"'{word}' : {count}")
```

OUTPUT

```
Enter a line : Welcome to programming, welcome to python.
‘welcome’ : 2
‘to’ : 2
‘programming’ : 1
‘python’ : 1


Enter a line : Hello World
‘hello’ : 1
‘world’ : 1
```

DATE : 08/10/2024

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

   PROGRAM

```
l=[int(i) for i in input("Enter List : ").split()]
new=['Over' if i>100 else i for i in l]
print("New List : ",new)
```

   OUTPUT

```
Enter List : 1 2 3 120 400 5 140
New List : [1, 2, 3, ‘Over’, ‘Over’, 5, ‘Over’]


Enter List : 5 600 700 8 800
New List : [5, ‘Over’, ‘Over’, 8, ‘Over’]
```

DATE : 08/10/2024

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

PROGRAM

```
import math

l=[i for i in input("Enter List : ").split()]
count=0
for i in l:
    count+= i.lower().count('a')
print("Count of Letter A : ",count)
```

OUTPUT

```
Enter List : Aswin, Neymar, Messi, Ronaldo
Count of Letter A : 4


Enter List : Pedri, Raphinha, Lewandowski
Count of Letter A : 3
```

DATE : 08/10/2024

7. **Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums.**

PROGRAM

```
l1=[int(i) for i in input("Enter List 1 : ").split()]
l2=[int(i) for i in input("Enter List 2 : ").split()]

#a
if len(l1) == len(l2):
   print("Length is same.")
else:
   print("Length is not same!")

#b
if sum(l1) == sum(l2) :
   print("Sum of Lists are equal.")
else:
   print("Sum is not equal!")

#c
c=set(l1).intersection(set(l2))
if len(c) != 0:
   print("Common Values : ",c)
else:
   print("No common elements!")
```

OUTPUT

```
Enter List 1 : 1 2 3 4
Enter List 2 : 3 4 5 6
Length is same.
Sum is not equal !
Common Values : {3, 4}

Enter List 1 : 1 2 3 4 5
Enter List 2 : 3 4 5 6
Length is not same!
Sum is not equal !
Common Values : {3, 4, 5}
```

DATE : 08/10/2024

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

PROGRAM

```
l=input("Enter a String : ")
f=l[0]

l1=l[1:].replace(f,'$')

print("New String : ",f+l1)
```

OUTPUT

```
Enter a String : Akash
New String : Ak$sh


Enter a String : Onion
New String : Oni$n
```

DATE : 10/10/2024

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

PROGRAM

```
s=input("Enter a String : ")
f=s[0]
l=s[-1:]

print("New String : ",l+s[1:-1]+f)
```

OUTPUT

```
Enter a String : Programming
New String : grogramminP


Enter a String : Python
New String : nythoP
```

DATE : 10/10/2024

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

PROGRAM

```
import math
r = int(input("Enter Radius : "))
area = math.pi * r * r
print("Area : ",round(area,2))
```

OUTPUT

```
Enter Radius : 5
Area : 78.54

Enter Radius : 6
Area :113.1
```

DATE : 10/10/2024

11. Find biggest of 3 numbers entered.

PROGRAM

```
n=[int(i) for i in input("Enter 3 Numbers separated by space : ").split()]
print("Biggest is : ",max(n))
```

OUTPUT

```
Enter 3 Numbers separated by space : 33 55 2
Biggest is : 55

Enter 3 Numbers separated by space : 5 8 2
Biggest is : 8
```

DATE : 10/10/2024

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

PROGRAM

```
file=input("Enter File Name : ")
temp=file.split(".")
ext= temp[-1] if len(temp) > 1 else ""
print("Extension : ",ext)
```

OUTPUT

```
Enter File Name : ABCD.img
Extension : img


Enter File Name : PQRS.docx
Extension : docx
```

DATE : 10/10/2024

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

PROGRAM

```
c=input("Enter Colors : ")
colors=[i.strip() for i in c.split(",")]
print("First Color : ",colors[0],"\nLast Color : ",colors[-1])
```

OUTPUT

```
Enter Colors : Red, Yellow, Blue
First Color : Red
Last Color : Blue


Enter Colors : White, Grey, Orange
Fisrt Color : White
Last Color : Orange
```

DATE : 15/10/2024

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

PROGRAM

```
x=int(input("Enter an Integer : "))
n1 = int(f"{x}") # nn
n2 = int(f"{x}{x}") # nn
n3 = int(f"{x}{x}{x}") #nnn
print(n1,"+",n2,"+",n3," = ",n1+n2+n3)
```

OUTPUT

```
Enter an Integer : 6
6 + 66 + 666 = 738


Enter an Integer : 8
8 + 88 + 888 = 984
```

DATE : 15/10/2024

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

PROGRAM

```
l1=[i for i in input("Enter List 1 : ").split()]
l2=[i for i in input("Enter List 2 : ").split()]
r=[i for i in l1 if i not in l2]
print("Result : ",r)
```

OUTPUT

```
Enter List 1 : Red, Yellow, Blue
Enter List 2 : Red, Green, Blue
Result : ['Yellow']
```

```
Enter List 1 : White, Yellow, Black
Enter List 2 : Red, Grey, Black
Result : ['White', 'Yellow']
```

DATE : 15/10/2024

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 :")
new1=s1[0]+s2[1]+s1[2:]
new2=s2[0]+s1[1]+s2[2:]
print("New String : “,new1,” ”,new2)
```

OUTPUT

```
Enter String 1 : Hello
Enter String 2 : World
New String : Hollo Werld
```

```
Enter String 1 : Python
Enter String 2 : Program
New String : Prthon Pyogram
```

DATE : 22/10/2024

17. Sort dictionary in ascending and descending order.

PROGRAM

```
d = {
    'apple': 3,
    'banana': 1,
    'cherry': 2,
    'date': 5,
    'blueberry': 4
}

#ascending
print("Ascending Order : ",dict(sorted(d.items())))

#descending
print("Descending Order : ",dict(sorted(d.items(),reverse=True)))
```

OUTPUT

```
Ascending Order : {‘apple’ : 3, ‘banana’: 1, ‘blueberry’ : 4, ‘cherry’ : 2, ‘date’ : 5}
Descending Order : {‘date’: 5, ‘cherry’ : 2, ‘blueberry’ : 4, ‘banana’ : 1, ‘apple’ :3}
```

DATE : 22/10/2024

18. Merge two dictionaries.

PROGRAM

```
d1 = {'a': 1, 'b': 2}
d2 = {'c': 3, 'd': 4}

merged = d1 | d2

print("Merged Dictionary : ", merged)
```

OUTPUT

Merged Dictionary : {‘a’ :1, ‘b’ : 2, ‘c’ : 3, ‘d’ : 4}

DATE : 22/10/2024

19. Find gcd of 2 numbers.

PROGRAM

```
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

num1 = int(input("Enter Num 1 : "))
num2 = int(input("Enter Num 2 :"))

result = gcd(num1, num2)
print("GCD : ",result)
```

OUTPUT

```
Enter Num 1 : 12
Enter Num 2 : 56
GCD : 4


Enter Num 1 : 68
Enter Num 2 : 54
GCD : 2
```

DATE : 22/10/2024

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

PROGRAM

```
oldList=[int(i) for i in input("Enter the numbers : ").split()]

newList=[i for i in oldList if i%2!=0]

print("List after removing even numbers :",newList)
```

OUTPUT

```
Enter the numbers : 1 2 3 4 5 6 7
List after removing even numbers : [1, 3, 5, 7]

Enter the numbers : 34 52 84 99 103
List after removing even numbers : [99, 103]
```

COURSE OUTCOME 2

DATE: 22/10/2024

1. **Program to find the factorial of a number.**

PROGRAM

```
import math
a=int(input("Enter a number : "))
print("Factorial : ",math.factorial(a))
```

OUTPUT

```
Enter a number : 5
Factorial : 120


Enter a number : 4
Factorial :24
```

DATE: 22/10/2024

2. **Generate Fibonacci series of N terms.**

PROGRAM

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

a, b = 0, 1
fibonacci_series = []
for i in range(n):
    fibonacci_series.append(a)
    a, b = b, a + b

print(f"Fibonacci series of {n} terms: {fibonacci_series}")
```

OUTPUT

```
Enter the number of terms : 8
Fibonacci series of 10 terms : [0, 1, 1, 2, 3, 5, 8, 13]


Enter the number of terms : 10
Fibonacci series of 10 terms : [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

DATE: 24/10/2024

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

PROGRAM

```
l=[int(i) for i in input("Enter List : ").split()]
print("Sum : ",sum(l))
```

OUTPUT

```
Enter Lis : 1 2 3 4 5
Sum : 15

Enter Lis : 1 2 3 4 5 6
Sum : 21
```

DATE: 24/10/2024

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

def is_all_digits_even(num):
    while num > 0:
        digit = num % 10
        if digit % 2 != 0:
            return False
        num //= 10
    return True

def even_digit_perfect_squares(start, end):
    even_digit_squares = []
    for num in range(start, end + 1):
        if num >= 1000 and num <= 9999 and is_all_digits_even(num):
            root = int(math.sqrt(num))
            if root * root == num:
                even_digit_squares.append(num)
    return even_digit_squares


start = 1000
end = 9999

even_digit_squares = even_digit_perfect_squares(start, end)
print(even_digit_squares)
```

OUTPUT

```
[4624, 6084, 6400, 8464]
```

DATE: 24/10/2024

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

OUTPUT

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

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

DATE: 24/10/2024

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

PROGRAM

```
from collections import Counter

s = input("Enter a string: ")

char_frequency = Counter(s)

print("Character Frequency:")
for char, freq in char_frequency.items():
    print(f"'{char}': {freq}")
```

OUTPUT

Enter a string : Hello World
‘H’ : 1
‘e’ : 1
‘l’ : 3
‘o’ : 2
‘W’ : 1
‘r’ : 1
‘d’ : 1

Enter a string : Programming
‘P’ : 1
‘r’ : 2
‘o’ : 1
‘g’ : 2
‘a’ : 1
‘m’ : 2
‘i’ : 1
‘n’ : 1

DATE: 24/10/2024

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 :")
a=s[-3:]

if a=='ing':
        print(s+'ly')
else:
        print(s+'ing')
```

OUTPUT

```
Enter a string : Dancing
Dancingly


Enter a string : Dance
Danceing
```

DATE: 29/10/2024

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

PROGRAM

```
s=[i for i in input("Enter some words :").split()]
print(len(max(s, key=len)))
```

OUTPUT

```
Enter some words : Python Programming
11

Enter some words : Hello Python
6
```

DATE: 29/10/2024

9. **Construct following pattern using nested loop**

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

PROGRAM

```
n = int(input("Enter the limit: "))

for i in range(n):
    for j in range(i + 1):
        print("*", end=" ")
    print()

for i in range(n):
    for j in range(n-i-1):
        print("*", end=" ")
    print()
```

OUTPUT

```
Enter the limit : 5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

DATE: 29/10/2024

10. Generate all factors of a number.

PROGRAM

```
def factors(a):
        for i in range(1,a+1):
                if a%i == 0:
                        print(i)


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

OUTPUT

```
Enter a number : 6
1
2
3
6


Enter a number : 7
1
7
```

DATE: 29/10/2024

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

PROGRAM

```
#Square
s=int(input("Enter Side of Square :"))
areaSq=lambda s:s*s
print("Area of Square = ",areaSq(s),"\n")


#Rectangle
l=int(input("Enter Length of Rectangle :"))
b=int(input("Enter Breadth of Rectangle :"))
areaRect= lambda l, b : l * b
print("Area of Rectangle = ",areaRect(l,b),"\n")


#Triangle
b=int(input("Enter Base of Triangle :"))
h=int(input("Enter Height of Triangle :"))
areaTri=lambda b,h: .5*b*h
print("Area of Triangle = ",areaTri(b,h),"\n")
```

OUTPUT

```
Enter Side of Square :5
Area of Square =  25

Enter Length of Rectangle :6
Enter Breadth of Rectangle :4
Area of Rectangle =  24

Enter Base of Triangle :3
Enter Height of Triangle :4
Area of Triangle =  6.0
```