

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 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.

DATE:08 /10/2024

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

PROGRAM

```
c = int(input("Enter Current Year : "))
e = int(input("Enter Ending Year : "))
print("Leap years are :")
for i in range(c,e):
    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 : 2030
Enter Ending Year : 2040
Leap years are :
2032 2036
```

```
Enter Current Year : 2020
Enter Ending Year : 2030
Leap years are :
2020 2024 2028
```

DATE: 10/10/2024

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

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

PROGRAM

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

OUTPUT

Enter list of integers : 1 -3 5 -7 8 9 -2

Positive Integers : [1, 5, 8, 9]

Enter list of integers : -5 5 4 -4 -3 3 2 -2 1 -1

Positive Integers : [5, 4, 3, 2, 1]

b. Square of N numbers

PROGRAM

```
list=[int(i) for i in input("Enter List : ").split()]
list1=[i*i for i in list]
print("Square of numbers :",list1)
```

OUTPUT

Enter List : 10 9 8 7 6

Square of numbers : [100, 81, 64, 49, 36]

Enter List : 1 2 3 4 5

Square of numbers : [1, 4, 9, 16, 25]

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

PROGRAM

```
word = input("Enter a word: ")
v = 'aeiouAEIOU'
vowel_list = [char for char in word if char in v]
print("Vowels in the word:", vowel_list)
```

OUTPUT

Enter a word: Hello world
Vowels in the word: ['e', 'o', 'o']

Enter a word: eutopia
Vowels in the word: ['e', 'u', 'o', 'i', 'a']

- d. List ordinal value of each element of a word

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:python
['p', 'y', 't', 'h', 'o', 'n']
List ordinal value of each element of a word:
[112, 121, 116, 104, 111, 110]

Enter a word:programming
['p', 'r', 'o', 'g', 'r', 'a', 'm', 'm', 'i', 'n', 'g']
List ordinal value of each element of a word:

[112, 114, 111, 103, 114, 97, 109, 109, 105, 110, 103]

DATE : 07-11-2024

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: hello world

Word occurrences:

hello: 1

world: 1

Enter a text: welcome to python

Word occurrences:

welcome: 1

to: 1

python: 1

DATE : 15-10-2024

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

PROGRAM

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

OUTPUT

Enter the integers:1 5 7 110 120

new list: [1, 5, 7, 'over', 'over']

Enter the integers:10 20 300 200 400

new list: [10, 20, 'over', 'over', 'over']

DATE : 10-10-2024

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

PROGRAM

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

OUTPUT

Enter a list of first names separated by spaces: anjana krish riya

The total occurrences of 'a': 4

Enter a list of first names separated by spaces: alan adeena tom

The total occurrences of 'a': 4

7. Enter 2 lists of integers. Check

a. Whether list are of same length

PROGRAM

```
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 1: 1 3 5 7

Enter values in list 2: 2 4 6 8

Length of list 1: 4

Length of list 2: 4

The list are same length of 4

Enter values in list 1: 10 20 30

Enter values in list 2: 15 45 60 75

Length of list 1: 3

Length of list 2: 4

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: 2 4 6 8

Enter values in list 2: 1 3 5 7

Sum of list1: 20

Sum of list2: 16

The sum are not equal

Enter values in list 1: 1 2 3 4 5

Enter values in list 2: 10 20 30

Sum of list1: 15

Sum of list2: 60

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: 1 2 3 4 5

Enter values in list 2: 2 4 6 8 10

Common values between the lists: {'2', '4'}

Enter values in list 1: 4 8 16 20

Enter values in list 2: 5 10 15

No common values

DATE : 24-10-2024

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("New String : ",s1)
```

OUTPUT

Enter a string:python program

New String : python \$rogra

Enter a string:restart

New String : resta\$

DATE : 24-10-2024

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

PROGRAM

```
s=input("Enter a string:")  
s1=s[-1]+s[1:-1]+s[0]  
print("String after first and last characters exchanged:",s1)
```

OUTPUT

Enter a string:Python

String after first and last characters exchanged: nythoP

Enter a string:programming

String after first and last characters exchanged: grogramminp

DATE : 01-10-2024

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

PROGRAM

```
r=float(input("Enter the radius:"))
pi=3.14
area=pi*r*r
print("Area of circle=",round(area,2))
```

OUTPUT

```
Enter the radius:6
Area of circle= 113.04
```

```
Enter the radius:10
Area of circle= 314.0
```

DATE : 01-10-2024

11. Find biggest of 3 numbers entered.

PROGRAM

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

OUTPUT

```
Enter 3 Numbers separated by space : 5 10 15\nlargest : 15
```

```
Enter 3 Numbers separated by space : 51 80 65\nlargest : 80
```

DATE : 24-10-2024

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 : image.jpg

Extension : jpg

Enter File Name : photo.jpeg

Extension : jpeg

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 :").split(",")  
l1 = [color.strip() for color in l1]  
print("Display first color:", l1[0])  
print("Display last color:", l1[-1])
```

OUTPUT

```
Enter list of colors :white,red,black  
Display first color: white  
Display last color: black
```

```
Enter list of colors :blue,green,pink  
Display first color: blue  
Display last color: pink
```

DATE : 03-10-2024

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:5

$n+nn+nnn = 5+55+555=615$

Enter value of n: 6

$n + nn + nnn = 6 + 66 + 666 = 738$

DATE : 15-10-2024

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

PROGRAM

```
l1 = input("Enter colors for List 1: ").split(",")
l2 = input("Enter colors for List 2: ").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

```
Enter colors for List 1: white,red,black
Enter colors for List 2: blue,white,pink,yellow
List 1: ['white', 'red', 'black']
List 2: ['blue', 'white', 'pink', 'yellow']
Print Colors in List 1 but not in List 2 as Set: {'black', 'red'}
Print as List: ['black', 'red']

Enter colors for List 1: red,blue,yellow
Enter colors for List 2: pink,blue,white
List 1: ['red', 'blue', 'yellow']
List 2: ['pink', 'blue', 'white']
Print Colors in List 1 but not in List 2 as Set: {'red', 'yellow'}
Print as List: ['red', 'yellow']
```

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("S1 After Swap : ",new1,"\nS2 After Swap : ",new2)
```

OUTPUT

```
Enter String 1 :python
Enter String 2 :programming
S1 After Swap : prthon
S2 After Swap : pyogramming
```

```
Enter String 1 : Hello
Enter String 2 : World
S1 After Swap : Hollo
S2 After Swap : Werld
```

DATE : 22-10-2024

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}
```

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 '**':",{**d1,**d2})
```

OUTPUT

```
Enter number of items for d1: 3
Enter key: kiwi
Enter value: 5
Enter key: orange
Enter value: 20
Enter key: apple
Enter value: 10
Enter number of items for d2: 2
Enter key: coffee
Enter value: 4
Enter key: tea
Enter value: 7
d1: {'kiwi': 5, 'orange': 20, 'apple': 10}
d2: {'coffee': 4, 'tea': 7}
After Update: {'kiwi': 5, 'orange': 20, 'apple': 10, 'coffee': 4, 'tea': 7}
Merging using '**': {'kiwi': 5, 'orange': 20, 'apple': 10, 'coffee': 4, 'tea': 7}
```

DATE : 29-10-2024

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 : 24  
Enter Num 2 :48  
GCD : 24
```

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

DATE : 07-11-2024

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 :1 2 3 4 5 6 7 8

List of integers after removing even nos.: [1, 3, 5, 7]

Enter the integers :35 88 100 55 95 40 74

List of integers after removing even nos.: [35, 55, 95]

COURSE OUTCOME 2

DATE : 03-10-2024

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:5

Factorial of 5 : 120

Enter a number:7

Factorial of 7 : 5040

DATE : 03-10-2024

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: 5
Fibonacci series of 5 terms: [0, 1, 1, 2, 3]
```

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

DATE : 08-10-2024

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:", sum)
```

OUTPUT

Enter the list elements separated by space: 1 2 3 4 5

List = [1,2,3,4,5]

Sum: 15

Enter the list elements separated by space: 5 10 15 20

List = [5,10,11,20,9]

Sum: 50

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: 4000

Enter the ending number: 9999

Numbers that are perfect squares and have all even digits: [4624, 6084, 6400, 8464]

Enter the starting number: 7000

Enter the ending number: 9000

Numbers that are perfect squares and have all even digits: [8464]

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:5

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
```

Enter a limit:6

```
1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
```

DATE : 24-10-2024

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: programming

```
p : 1
r : 2
o : 1
g : 2
a : 1
m : 2
i : 1
n : 1
```

Enter a string: hello

```
h: 1
e: 1
l: 2
o: 1
```

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 :play
Playing
```

```
Enter a string :dancing
Dancingly
```

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 :hello world

5

Enter some words : Python Programming

11

DATE : 01-10-2024

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:4

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

* *

*

Enter a limit:5

*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*

DATE : 29-10-2024

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: 10

Factors of 10 :

1
2
5
10

Factors of : 3

1
3

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 :4
Area of Square = 16
```

```
Enter Length of Rectangle :5
Enter Breadth of Rectangle :4
Area of Rectangle = 20
```

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

```
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 :2

Enter Height of Triangle :6

Area of Triangle = 6.0