

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:

a. 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

b. 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

c. 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.

d. 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

e. 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.

f. 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

g. 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 analyze 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 : 18-09-2023

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

PROGRAM

```
cy=int(input("Enter the current year : "))
fy=int(input("Enter the ending year : "))
print("Leap years between",cy,"and",fy,"are : ",end=" ")
for i in range(cy,fy):
    if (i%400==0) and (i%100==0):
        print(i)
    elif (i%4==0) and (i%100!=0):
        print(i,end=" ")
```

OUTPUT

Enter the current year : 2020

Enter the ending year : 2030

Leap years between 2020 and 2030 are : 2020 2024 2028

DATE : 18-09-2023

3. List comprehensions:

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

PROGRAM

```
l=[int(i) for i in input("Enter elements : ").split()]
l_p=[i for i in l if i>=0]
print("List of +ve numbers : ",l_p)
```

OUTPUT

```
Enter elements : -5 -4 -3 -2 -1 0 1 2 3 4 5
List of +ve numbers : [0, 1, 2, 3, 4, 5]
```

- b. Square of N numbers.

PROGRAM

```
n=int(input("Enter no of terms : "))
sq_list=[i*i for i in range(n)]
print("List of square of N numbers : ",sq_list)
```

OUTPUT

```
Enter no of terms : 5
List of square of N numbers : [0, 1, 4, 9, 16]
```

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

PROGRAM

```
c=input("Enter a word : ")
v_list=[i for i in c if i in "AEIOUaeiou"]
print("List of vowels : ",v_list)
```

OUTPUT

```
Enter a word : Hello
List of vowels : ['e', 'o']
```

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

PROGRAM

```
x=input("Enter a word : ")
o_list=[ord(i) for i in x]
print("List of ordinal values : ",o_list)
```

OUTPUT

```
Enter a word : Hello
List of ordinal values : [72, 101, 108, 108, 111]
```


DATE : 18-09-2023

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

PROGRAM

```
str=input("Enter a line : ")
l=[]
l=str.split();
wc=[l.count(i) for i in l]
print("Occurrences = ",dict(zip(l,wc)))
```

OUTPUT

```
Enter a line : hello world this is python programming python is simple
and powerful
Occurrences = {'hello': 1, 'world': 1, 'this': 1, 'is': 2, 'python': 2,
'programming': 1, 'simple': 1, 'and': 1, 'powerful': 1}
```

DATE : 18-09-2023

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 elements : ").split()]
new_l=['over' if i>100 else i for i in l]
print("Result : ",new_l)
```

OUTPUT

```
Enter elements : 50 100 150 200 75
Result : [50, 100, 'over', 'over', 75]
```

DATE : 20-09-2023

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

PROGRAM

```
names=[i for i in input("Enter the names : ").split()]
sm_names=[i.lower() for i in names]
for i in sm_names:
    print("'a' repeats in ",i,i.count('a')," times")
```

OUTPUT

```
Enter the names : Anirudh Suraj Vishwesh Shivani Srinidhi
'a' repeats in anirudh 1 times
'a' repeats in suraj 1 times
'a' repeats in vishwesh 0 times
'a' repeats in shivani 1 times
'a' repeats in srinidhi 0 times
```

DATE : 20-09-2023

7. Enter 2 lists of integers. Check
- Whether lists are of the same length.
 - Whether the list sums to the same value.
 - Whether any value occurs in both.

PROGRAM

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

if len(l1)==len(l2):
    print("Lists are of same length")
else:
    print("Lists are of different length")

if sum(l1)==sum(l2):
    print("Sum of lists are of same")
else:
    print("Sum of lists are different")

res=[i for i in l1 if i in l2]
print("Common elements : ",res)
```

OUTPUT

```
Enter list 1 elements : 10 20 30 40 50
Enter list 1 elements : 10 25 30 45 50
Lists are of same length
Sum of lists are different
Common elements : [10, 30, 50]
```

DATE : 20-09-2023

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

PROGRAM

```
str1=input("Enter a word : ")
f_c=str1[0]
str1=str1.replace(f_c,'$')
str1=f_c+str1[1:]
print("Replaced string : ",str1)
```

OUTPUT

```
Enter a word : anirudhjbhatt
Replaced string : anirudhjbh$tt
```

DATE : 20-09-2023

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

PROGRAM

```
name=input("Enter a name : ")  
print("Result :",name[-1]+name[1:-1]+name[0])
```

OUTPUT

```
Enter a name : Anirudh  
Result : hnirudA
```

DATE : 20-09-2023

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

PROGRAM

```
r=float(input("Enter the radius : "))  
print("Area : ",3.14*r*r)
```

OUTPUT

```
Enter the radius : 100  
Area : 31400.0
```

DATE : 27-09-2023

11. Find the biggest of 3 numbers entered.

PROGRAM

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

OUTPUT

```
Enter the first number : 10
Enter the second number : 20
Enter the third number : 30
30 is the largest number
```

```
Enter the first number : 10
Enter the second number : 40
Enter the third number : 30
40 is the largest number
```

```
Enter the first number : 50
Enter the second number : 20
Enter the third number : 30
50 is the largest number
```


DATE : 27-09-2023

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

PROGRAM

```
f_name=input("Enter a filename : ")  
f_ext=f_name.split(".")  
print("Extension : ",f_ext[-1])
```

OUTPUT

```
Enter a filename : PGM12.py  
Extension : py
```

DATE : 27-09-2023

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

PROGRAM

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

OUTPUT

```
Enter the colors : Red Yellow Green Blue Violet  
First color : Red  
Last color : Violet
```

DATE : 27-09-2023

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

PROGRAM

```
n=int(input("Enter a number : "))  
print("Result :",n+n*11+n*111)
```

OUTPUT

Enter a number : 5

Result : 615

DATE : 27-10-2023

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

PROGRAM

```
c1=[i for i in input("Enter the colors list 1 : ").split()]
c2=[i for i in input("Enter the colors list 2: ").split()]
res=[i for i in c2 if i not in c1]
print("Colors from color-list 1 not in colo-list 2 : ",res)
```

OUTPUT

```
Enter the colors list 1 : Red Yellow Green Blue Violet
Enter the colors list 2: Red Pink Green Magenta Violet
Colors from color-list 1 not in colo-list 2 : ['Pink', 'Magenta']
```

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 : Hello
Enter string 2 : World
Swapped string : Wello Horld
```

DATE : 04-10-2023

17. Sort the dictionary in ascending and descending order.

PROGRAM

```
d = {'Anirudh': 10, 'Shivani': 20, 'Srinidhi': 30, 'Aryan': 40, 'Suraj': 50}
print("Ascending order : ", dict(sorted(d.items())))
print("Descending order : ", dict(sorted(d.items(), reverse=True)))
```

OUTPUT

Ascending order : {'Anirudh': 10, 'Aryan': 40, 'Shivani': 20, 'Srinidhi': 30, 'Suraj': 50}

Descending order : {'Suraj': 50, 'Srinidhi': 30, 'Shivani': 20, 'Aryan': 40, 'Anirudh': 10}

DATE : 04-10-2023

18. Merge two dictionaries.

PROGRAM

```
d1 = {'Anirudh': 10, 'Vishwesh': 20, 'Suraj': 30}
d2 = {'Shivani': 20, 'Srinidhi': 40, 'Saubhagya': 50}
print("Merged dictionaries :", d1|d2)
```

OUTPUT

Merged dictionaries : {'Anirudh': 10, 'Vishwesh': 20, 'Suraj': 30, 'Shivani': 20, 'Srinidhi': 40, 'Saubhagya': 50}

DATE : 04-10-2023

19. Find gcd of 2 numbers.

PROGRAM

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

OUTPUT

```
Enter the first number : 10
Enter the second number : 20
GCD : 10
```


DATE : 04-10-2023

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

PROGRAM

```
l=[int(i) for i in input("Enter integers : ").split()]  
l=[i for i in l if i%2!=0]  
print("List after removal of even numbers : ",l)
```

OUTPUT

Enter integers : 1 2 3 4 5 6 7 8 9 10

List after removal of even numbers : [1, 3, 5, 7, 9]

COURSE OUTCOME 2

DATE : 09-10-2023

1. Program to find the factorial of a number.

PROGRAM

```
def fact(x):  
    if x==1:  
        return 1  
    else:  
        return x*fact(x-1)  
x=int(input("Enter the number : "))  
print("Factorial : ",fact(x))
```

OUTPUT

```
Enter the number : 5  
Factorial : 120
```

DATE : 09-10-2023

2. Generate Fibonacci series of N terms.

PROGRAM

```
n=int(input("Enter the no of terms : "))
a=0
b=1
c=a+b
print("Fibonacci series : ",a,b,c,end=" ")
for i in range(3,n):
    a=b
    b=c
    c=a+b
    print(c,end=" ")
```

OUTPUT

```
Enter the no of terms : 5
Fibonacci series : 0 1 1 2 3
```

DATE : 09-10-2023

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

PROGRAM

```
l=[int(i) for i in input("Enter elements : ").split()]\nprint("Sum of elements : ",sum(l))
```

OUTPUT

Enter elements : 10 20 30 40 50

Sum of elements : 150

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 the step number accepted from the user.

Eg: N=4

1

2 4

3 6 9

4 8 12 16

PROGRAM

```
n=int(input("Enter the no of rows : "))
for i in range(1,n+1):
    for j in range(1,i+1):
        print(i*j,end=" ")
    print()
```

OUTPUT

Enter the no 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

```
str1=input("Enter a string : ")  
wc=[str1.count(i) for i in str1]  
print(dict(zip(str1,wc)))
```

OUTPUT

```
Enter a string : Muthoot Institute of Technology and Science  
{ 'M': 1, 'u': 2, 't': 5, 'h': 2, 'o': 5, ' ': 5, 'I': 1, 'n': 4, 's': 1, 'i': 2, 'e': 4, 'f': 1,  
'T': 1, 'c': 3, 'l': 1, 'g': 1, 'y': 1, 'a': 1, 'd': 1, 'S': 1 }
```

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("String :",str2)
```

OUTPUT

```
Enter a string : String
String : Stringly
```


DATE : 11-10-2023

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

PROGRAM

```
l=[i for i in input("Enter the words : ").split()]

max=len(l[0])
temp=l[0]
for i in l:
    if len(i)>max:
        max=len(i)
        temp=i
print("Word with max length is",temp,"It's length is",max)
)
```

OUTPUT

Enter the words : Anirudh Vishwesh Suraj Shivani Srinidhi
Word with max length is Vishwesh It's length is 8

DATE : 11-10-2023

9. Construct following pattern using nested loop

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

PROGRAM

```
n=int(input("Enter the no of rows : "))
for i in range(1,n+1):
    print('*'*i)
for i in range(n-1,0,-1):
    print('*'*i)
```

OUTPUT

```
Enter the no of rows : 5
*
**
***
****
*****
****
***
**
*
```

DATE : 11-10-2023

10. Generate all factors of a number.

PROGRAM

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

OUTPUT

Enter a number : 5

Factors of 5 are : 1 5

DATE : 11-10-2023

11. Write lambda functions to find the 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 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))

b=int(input("Enter the base of triangle : "))
h=int(input("Enter the height of triangle : "))
print("Area of triangle : ",area3(b,h))
```

OUTPUT

```
Enter the side of square : 5
Area of square : 25
Enter the length of rectangle : 10
Enter the breadth of rectangle : 20
Area of rectangle : 200
Enter the base of triangle : 10
Enter the height of triangle : 20
Area of triangle : 100.0
```