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

```
start = int(input("Enter start year: "))
end = int(input("Enter end year: "))
if a < end:
  print ("the list of leap year" + str(start) + " and " + str(end)+ ":")
  while start < end:
    if start % 4 == 0 and start % 100 != 0:
        print(start)
    if start % 100 == 0 and start % 400 == 0:
        print(start)
    start += 1
```

OUTPUT

Enter start year: 2032

Enter end year: 2036

the list of leap year 2032 and 2036:

2024

DATE : 18-09-2023

3. List comprehensions:
    a. Generate a positive list of numbers from a given list of integers.

PROGRAM

```
l1=[1,-1,-2,2,-3,3,4,5,-5,6,7]
l2=[i for i in l1 if i>0]
print("Positive integers:",l2)
```

OUTPUT

Positive integers: [1, 2, 3, 4, 5, 6, 7]

    b. Square of N numbers.

PROGRAM

```
n=int(input("Enter limit:"))
l=[i*i for i in range (n)]
l
```

OUTPUT

Enter limit:6
[0,1,4,9,16,25]

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

PROGRAM

```
a=input("Enter a word:")
v=[i for i in a if i in 'aeiouAEIOU']
v
```

OUTPUT

```
Enter a word:education
['e', 'u', 'a', 'i', 'o']
```

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

PROGRAM

```
n=int(input("Enter limit:"))
l=[i*i for i in range (n)]
l
```

OUTPUT

```
Enter limit:5
[0, 1, 4, 9, 16]
```

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(dict(zip(l,wc)))
```

OUTPUT

Enter a line: Karthik is a musician

{'Karthik': 1, 'is': 1, 'a': 1, 'musician': 1}

DATE : 18-09-2023

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

PROGRAM

```
list=[]
n=int(input("enter the number:"))
for i in range (0, n):
nu=int(input())
  if nu>100:
  list.append('over')
  else:
     list.append(nu)
print(list)
```

OUTPUT

```
enter the number:5
6
8
106
125
19
[6, 8, 'over', 'over', 19]
```

DATE : 20-09-2023

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

PROGRAM

```
name=['basil', 'ashwin']for
i in name:
        print("a","occurs in",i,i.count('a'),"times")
```

OUTPUT

a occurs in basil 1 times

a occurs in ashwin 1

times

DATE : 20-09-2023

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

PROGRAM

```
l1=[]
l2=[]
n1=int(input("Enter the size of list 1 : "))
print("Enter list 1 : ")
for i in range(n1):
  i=int(input())
  l1.append(i)

n2=int(input("Enter the size of list 2 : "))
print("Enter list 2 : ")
for i in range(n2):
  i=int(input())
  l2.append(i)

if n1==n2:
  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 the size of list 1 : 3
Enter list 1 :
2
4
6
Enter the size of list 2 : 3
Enter list 2 :
2
3
7
Lists are of same length
Sum of lists are different
Common elements : [2]
```

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(str1)
```

OUTPUT

```
Enter a word :onion
oni$n
```

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 : ")
name[-1]+name[1:-1]+name[0]
```

OUTPUT

Enter a name : python

'nythop’

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 : 30

Area : 2826

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 : 1
Enter the second number : 2
Enter the third number : 3
3 is the largest number

Enter the first number : 23
Enter the second number :4
Enter the third number :31
31 is the largest number

Enter the first number : 6
Enter the second number : 9
Enter the third number : 3
9 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 : index.html
Extension : html
```

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

```
color_input = input("Enter a list of colors separated by commas: ")

colors = color_input.split(',')


colors = [color.strip() for color in colors]


if len(colors) >= 2:

    print(f"First color: {colors[0]}")
    print(f"Last color: {colors[-1]}")
else:
    print("Please enter at least two colors separated by commas.")
```

OUTPUT

Enter a list of colors separated by commas red,blue,violet,green

First color: red

Last color: green

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=["Red","Black","Green","Blue","Violet","Purple"]
c2=["Cyan","Black","Pink","Blue","Violet","Brown"]
res=[i for i in c2 if i not in c1]
print("Colors from color-list 1 not in colo-list 2 : ",res)
```

OUTPUT

Colors from color-list 1 not in colo-list 2 : ['Cyan', 'Pink' ,'Brown']

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 :zig
Enter string 2 : zag
Swapped string : zag zig
```

DATE : 04-10-2023

17. Sort the dictionary in ascending and descending order.

PROGRAM

```
d = {'ashwin': 8, 'beema': 10,'basil': 15}
 print("Ascending order : ",dict(sorted(d.items())))
print("Descending order : ",dict(sorted(d.items(), reverse=True)))
```

OUTPUT

```
Ascending order   : {'ashwin': 8, ' basil  ': 15, 'beema': 10}
Descending order : {'beema': 10, ' basil ': 15, ' ashwin ': 8}
```

DATE : 04-10-2023

18. Merge two dictionaries.

PROGRAM

```
d1 = {'Anu': 10,'Manu': 20, 'Ram': 30}
d2 = {'jeevan': 20,'hanna': 40, 'manna': 50}
print("Merged dictionaries :",d1|d2)
```

OUTPUT

```
Merged dictionaries : {'Anu': 10, 'Manu': 20, 'Ram': 30, 'jeevan': 20,
'hanna': 40, 'manna': 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 : 50

Enter the second number :80

GCD : 10

DATE : 04-10-2023

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

PROGRAM

```
l=[]
n=int(input("Enter the size of the list : "))
print("Enter elements : ")
for i in range(n):
  i=int(input())
  l.append(i)

for i in l:
  if i%2==0:
    l.remove(i)

print("List after removal of even numbers : ",l)
```

OUTPUT

```
Enter the size of the list : 5
Enter elements :
1
2
3
4
5
List after removal of even numbers : [1, 3, 5]
```