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 start < 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: 2000
Enter end year: 2025
the list of leap year2000 and 2025:
2000
2004
2008
2012
2016
2020
2024
```

DATE : 18-09-2023

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

PROGRAM

```
num=[1,2,3,4,5,-1,-2,-3]
num1=[i for i in num if i>0]
print("positive numbers is",num1)
```

OUTPUT

```
positive numbers is [1, 2, 3, 4, 5]
```

    b. Square of N numbers.

PROGRAM

```
n=int(input("Enter the Number:"))
num=[i*i for i in range(n)]
num
```

OUTPUT

Enter the Number:4
[0, 1, 4, 9]

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

PROGRAM

```
vow=input("enter the word:")
vow1=[i for i in vow if i in 'aeiouAEIOU']
vow1
```

OUTPUT

```
enter the word:adfg
['a']
```

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

PROGRAM

```
word=input("enter a word:")
result=[ord(i) for i in word]
result
```

OUTPUT

```
enter a word:rgfw
[114, 103, 102, 119]
```

DATE : 18-09-2023

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

PROGRAM

```
name=["hello hello hello hello world"]
for i in name:
  print("hello","occurs in",i.count("hello"))
  print("world","occurs in",i.count("world"))
```

OUTPUT

```
hello occurs in 4
world occurs in 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 number3
1
2
202
[1, 2, 'over']
```

DATE : 20-09-2023

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

PROGRAM

```
name=['eren:', 'sukuno:', 'luffy:', 'goku:']
for i in name:
  print("u","occurs in",i,i.count('u'))
```

OUTPUT

```
u occurs in eren: 0
u occurs in sukuno: 2
u occurs in luffy: 1
u occurs in goku: 1
```

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

```
num1=[1,23,44]
num2=[22,44,55]
len1=len(num1)
len2=len(num2)
if len1==len2:
  print("length are same")
else:
  print("length are not same")
```

OUTPUT

```
length are same
```

PROGRAM

```
num1=[1,23,44]
num2=[22,44,55]
sum=0
sum1=0
for i in num1:
  sum=sum+i
print(sum)
for i in num2:
  sum1=sum1+i
print(sum1)
if sum==sum1:
  print("sum are equal")
else:
  print("sum are not equal")
```

OUTPUT

```
68
121
sum are not equal
```

PROGRAM

```
num1=[2,5,3]
num2=[6,4,5,1]
for i in num1:
  for j in num2:
    if i==j:
      print("true")
      break;
```

OUTPUT

```
true
```

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

```
word=(input("Enter a word:"))
wrd=word[0]
str=word.replace(wrd,'$')
str=wrd+str[1:]
print(str)
```

OUTPUT

```
Enter a word:yuyyy
yu$$$
```

DATE : 20-09-2023

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

PROGRAM

```
word=input("enter a word:")
word[-1]+word[1:-1]+word[0]
```

OUTPUT

```
enter a word:word
'dorw'
```

DATE : 20-09-2023

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

PROGRAM

```
r=int(input("enter the radius:"))
area=3.14*r*r
print(area)
```

OUTPUT

```
enter the radius:4
50.24
```

DATE : 27-09-2023

11. Find the biggest of 3 numbers entered.

PROGRAM

```
a=int(input("Enter the number: "))
b=int(input("Enter the second number:"))
c=int(input("enter the third number:"))
if a>b and a>c:
  print(a)
elif (b>a and b>c):
  print(b)
elif (c>a and c>b):
  print(c)
```

OUTPUT

```
Enter the number: 4
Enter the second number:5
enter the third number:6
6
```

DATE : 27-09-2023

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

PROGRAM

```
name=input("Enter a filename : ")
name1=name.split(".")
print("Extension : ",name1[-1])
```

OUTPUT

```
Enter a filename : trav.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

```
colors=['black', 'green', 'blue', 'red']
print(colors[0],colors[-1])
```

OUTPUT

```
black red
```

PROGRAM

```
colors=[]
for i in range(4):
  color=input("enter the colors:")
  colors.append(color)
print(colors)
print(colors[0],colors[-1])
```

OUTPUT

```
enter the colors:red
enter the colors:blue
enter the colors:black
enter the colors:green
['red', 'blue', 'black ', 'green']
red green
```

DATE : 27-09-2023

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

PROGRAM

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

OUTPUT

```
Enter a number : 4
      Result : 492
```

DATE : 27-10-2023

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

PROGRAM

```
color1 = ["Red", "Green", "Blue", "Yellow", "Orange"]
color2 = ["Black", "White", "Green", "Yellow"]
list = [color for color in color1 if color not in color2]
print(list)
```

OUTPUT

```
['Red', 'Blue', 'Orange']
```

DATE : 04-10-2023

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

PROGRAM

```
string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")
line = string2[0] + string1[1:]
line2 = string1[0] + string2[1:]
result = line + " " + line2
print("Resulting string after swapping characters at position 1:", result)
```

OUTPUT

```
Enter the first string: Christo
Enter the second string: Thomas
Resulting string after swapping characters at position 1: Thristo Chomas
```

DATE : 04-10-2023

17. 17. Sort the dictionary in ascending and descending order.

PROGRAM

```
Fruits = {'orange': 3, 'apple': 2, 'mango': 5, 'date': 1}
ascending = dict(sorted(Fruits.items(), key=lambda item: item[1]))
descending = dict(sorted(Fruits.items(), key=lambda item: item[1],
reverse=True))
print("Ascending order:", ascending)
print("Descending order:", descending)
```

OUTPUT

```
Ascending order: {'date': 1, 'apple': 2, 'orange': 3, 'mango':5}
Descending order: {'mango': 5, 'orange': 3, 'apple': 2, 'date': 1}
```

DATE : 04-10-2023

18. Merge two dictionaries.

PROGRAM

```
d1={"orange":1,"apple":2}
d2={"mango":3,"guava":4}
m={**d1,**d2}
m
```

OUTPUT

```
{'orange': 1, 'apple': 2, 'mango': 3, 'guava': 4}
```

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 : 4
Enter the second number : 5
GCD :  1
```

DATE : 04-10-2023

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

PROGRAM

```
list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
list2 = [num for num in list1 if num % 2 != 0]
print("removed even numbers",list2)
```

OUTPUT

```
removed even numbers [1, 3, 5, 7, 9]
```

COURSE OUTCOME 2

DATE : 09-10-2023

1. Program to find the factorial of a number.

PROGRAM

```
n=int(input("enter the number:"))
f=1
for i in range(1, n+1):
 f=i*f
print(f)
```

OUTPUT

```
Enter a number: 4
24
```

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(a)
print(b)
print(c)
for i in range(3,n):
  a=b
  b=c
  c=a+b
  print(c)
```

OUTPUT

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

DATE : 09-10-2023

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

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)

print("Sum : ",sum(l))
```

OUTPUT

```
Enter the size of the list : 3
Enter elements :
4
5
6
Sum :  15
```

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

```
num = int(input("Enter the number of steps for the pyramid: "))

for i in range(1, num + 1):
    for j in range(1, i + 1):

        value = i * j

        print(value, end=" ")

    print()
```

OUTPUT

```
Enter the number of steps for the pyramid: 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

```
input_string = input("Enter a string: ")
char_count = {}
for char in input_string:
    char_count[char] = char_count.get(char, 0) + 1
for char, count in char_count.items():
    print(f"'{char}': {count}")
```

OUTPUT

```
Enter a string: Christo
'C': 1
'h': 1
'r': 1
'i': 1
's': 1
't': 1
'o': 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

```
s=input("Enter string:")
if s.endswith("ing"):
  b=s+'ly'
else:
  b=s+"ing"
print(b)
```

OUTPUT

```
Enter string:punch
punching
```

DATE : 11-10-2023

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

PROGRAM

```
num = int(input("Enter the size: "))
num1 = [input("Enter word: ") for _ in range(num)]
temp = max(num1, key=len)
print("Word with max length is", temp, "Its length is", len(temp))
```

OUTPUT

```
Enter the size: 3
Enter word: christo
Enter word: football
Enter word: timetravel
Word with max length is timetravel Its length is 10
```

DATE : 11-10-2023

9. Construct following pattern using nested loop

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

PROGRAM

```
n=int(input("enter n"))
for i in range(1,n+1):
  print ('*'*i)
for i in range(n-1,0,-1):
  print ('*'*i)
```

OUTPUT

```
enter n4
*
**
***
****
***
**
*
```

DATE : 11-10-2023

10. Generate all factors of a number.

PROGRAM

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

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

```
len=int(input("enter the length:"))
side=int(input("enter the side:"))
bre=int(input("enter the breadth"))
hei=int(input("enter the height"))
rec=lambda len, hei:len*hei
squ=lambda side:side*side
tri=lambda bre, hei:.5* bre*hei
print("area of rectangle:",rec(len,hei))
print("area of square:", squ(side))
print("area of triangle:", tri(bre,hei))
```

OUTPUT

```
enter the length:4
enter the side:5
enter the breadth6
enter the height7
area of rectangle: 28
area of square: 25
area of triangle: 21.0
```