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

- 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 Sonar Source for continuous inspection of code quality to perform fully automated code reviews / analysis to detect code smells, bugs, performance enhancements and security vulnerabilities.

DATE: 04/10/2024

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

PROGRAM

```
year1=int(input("Enter starting year "))
year2=int(input("Enter final year "))
for x in range(year1,year2):
  if (x%4==0 and x%100!=0 )or x%400==0:
    print("Leap year ",x)
```

OUTPUT

```
Enter starting year 2025
Enter final year 2040
Leap year  2028
Leap year  2032
Leap year  2036

Enter starting year 2020
Enter final year 2030
Leap year  2020
Leap year  2024
Leap year  2028
```

DATE: 03/10/2024

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

PROGRAM

```
l=input("Enter list of integers seperated by spaces:")
l1=[int(num) for num in l.split()]
pl=[num for num in l1 if num>0]
print("List of positive numbers: ",pl)
```

OUTPUT

```
Enter list of integers separated by spaces: 2 3 5 -7 8 -11 4
List of positive numbers:  [2, 3, 5, 8, 4]

Enter list of integers separated by spaces 2 5 -6 2 -7 9
List of positive numbers:  [2, 5, 2, 9]
```

(b).Square of N numbers

PROGRAM

```
l=input("Enter list of integers separated by spaces: ")
l1=[int(num) for num in l.split()]
print("Square of numbers:")
l2=[(num*num) for num in l1]
print(l2)
```

OUTPUT

```
Enter list of integers separated by spaces: 2 3 4 5
Square of numbers:
[4, 9, 16, 25]

Enter list of integers separated by spaces: 5 6 7 8 9
Square of numbers:
[25, 36, 49, 64,81]
```

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

PROGRAM

```
l=input("Enter a word: ")
l1=[x for x in l]
print(l1)
print("Vowels:")
l2=["a","e","i","o","u","A","E","I","O","U"]
l3=[x for x in l1 if x in l2]
print(l3)
```

OUTPUT

```
Enter a word: english
['e', 'n', 'g', 'l', 'i', 's', 'h']
Vowels:
['e', 'i']

Enter a word: malayalam
['m', 'a', 'l', 'a', 'y', 'a', 'l', 'a', 'm']
Vowels:
['a', 'a', 'a', 'a']
```

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

PROGRAM

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

OUTPUT

```
Enter a word : apple
The ordinal values of the characters in the word 'apple' : [97, 112, 112, 108, 101]

Enter a word : grapes
The ordinal values of the characters in the word 'grapes' : [103, 114, 97, 112,101,115]
```

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 : grapes orange grapes apple
'apple': 1
'orange': 1
'grapes': 2

Enter a line : banana orange mango orange mango
'banana': 1
'orange': 2
'mango': 2
```

DATE: 08/10/2024

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

PROGRAM

```
u=input("Enter list of integers seperated by spaces: ")
numbers=u.split()
#numbers=[int(num) for num in u.split()]
result=[]
for num in numbers:
  numbers=int(num)
  if numbers > 100:
    result.append("over")
  else:
    result.append(numbers)
print("new list")
print(result)
```

OUTPUT

```
Enter list of integers separated by spaces 40 20 41 160 8 400 55
New list
[40, 20, 41, 'over', 8, 'over', 55]

Enter list of integers separated by spaces 20 30 160 77 180 55 400
New list
[20, 30, 'over', 77, 'over', 55, '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 : Ann
Count of Letter A :  1

Enter List : Anna
Count of Letter A : 2
```

DATE: 08/10/2024

7. Enter 2 lists of integers. Check
   (a) Whether list is of same length
   (b) whether list sums to same value
   (c) whether any value occur in both

PROGRAM

```
l1=[int(i) for i in input("Enter List 1 : ").split()]
l2=[int(i) for i in input("Enter List 2 : ").split()]
if len(l1) == len(l2):
   print("Length is same.")
else:
   print("Length is not same!")
if sum(l1) == sum(l2) :
   print("Sum of Lists are equal.")
else:
   print("Sum is not equal!")
c=set(l1).intersection(set(l2))
if len(c) != 0:
   print("Values : ",c)
else:
   print("No common elements!")
```

OUTPUT

```
Enter List 1 : 4 5 6 7 8 9
Enter List 2 : 4 5 7
Length is not same!
Sum is not equal!
Values :  {4, 5, 7}

Enter List 1 : 4 5 6 7 8
Enter List 2 : 8 7 6 5 4
Length is same.
Sum of Lists are equal.
Values :  {4, 5, 6, 7, 8}
```

DATE: 08/10/2024

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

PROGRAM

```
l=input("Enter a String : ")
f=l[0]
l1=l[1:].replace(f,'$')
print("New String : ",f+l1)
```

OUTPUT

```
Enter a String : malayalam
New String :  malayala$

Enter a String : tomato
New String :  toma$o
```

DATE: 10/10/2024

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

PROGRAM

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

OUTPUT

```
Enter a String : welcome
New String :  eelcomw

Enter a String : update
New String :  epdatu
```

DATE: 10/10/2024

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

PROGRAM

```
r=int(input("Enter radius: "))
a=3.14*r*r
print("Area of circle: ",a)
```

OUTPUT

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

Enter radius: 20
Area of circle:  1256.0
```

DATE: 10/10/2024

11. Find biggest of 3 numbers entered

PROGRAM

```
a=int(input("Enter num 1: "))
b=int(input("Enter num 2:"))
c=int(input("Enter num 3: "))
if a > b and a > c:
  print(a ,"is greater")
elif b > a and b > c:
  print(b ,"is greater")
elif c > a and c > b:
  print(c ,"is greater")
else:
  print("all are equal")
```

OUTPUT

```
Enter num 1: 35
Enter num 2: 15
Enter num 3: 5
35 is greater

Enter num 1: 20
Enter num 2: 50
Enter num 3: 30
50 is greater

Enter num 1: 10
Enter num 2: 20
Enter num 3: 30
30 is greater
```

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 : file.txt
Extension :  txt

Enter File Name : lulu.jpg
Extension :  jpg
```

DATE: 10/10/2024

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

PROGRAM

```
l1=[i for i in input("Enter the colors in list1: ").split()]
print("List:")
print(l1)
print("first color: ",l1[1])
print("last color: ",l1[-1])
```

OUTPUT

```
Enter the colors in list1: orange  green blue red
List:
['orange', 'green', 'blue', 'red']
first color:  orange
last color:  red

Enter the colors in list1: blue red yellow black
List:
['blue', 'red', 'yellow', 'black']
first color:  blue
last color:  black
```

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}")
n2 = int(f"{x}{x}")
n3 = int(f"{x}{x}{x}")
print(n1,"+",n2,"+",n3," = ",n1+n2+n3)
```

OUTPUT

```
Enter an Integer : 5
5 + 55 + 555  =  615

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

DATE: 15/10/2024

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

PROGRAM

```
list1=[i for i in input("Enter the colors in list1: ").split()]
list2=[i for i in input("Enter the colors in list2: ").split()]
result=[i for i in list1 if i not in list2]
print("Colors in list1 not in list2 ",result)
```

OUTPUT

```
Enter the colors in list1: orange apple pineapple blueberry grapes
Enter the colors in list2: apple orange banana
Colors in list1 not in list2  ['pineapple', 'blueberry', 'grapes']

Enter the colors in list1: black blue yellow orange
Enter the colors in list2: yellow pink orange
Colors in list1 not in list2  ['black', 'blue']
```

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

OUTPUT

```
Enter String 1 :mango
Enter String 2 :orange
S1 After Swap :  mrngo
S2 After Swap :  oaange

Enter String 1 :car
Enter String 2 :bike
S1 After Swap :  cir
S2 After Swap :  bake
```

DATE: 15/10/2024

17. Sort dictionary in ascending and descending order.

PROGRAM

```
d={"apple":10,"orange":20,"banana":5,"kiwi":2}
print("Dictionary ",d)
aresult=dict(sorted(d.items()))
dresult=dict(sorted(d.items(),reverse=True))
print("Dictionary in ascending order ",aresult)
print("Dictionary in descending order ",dresult)
```

OUTPUT

```
Dictionary  {'apple': 10, 'orange': 20, 'banana': 5, 'kiwi': 2}
Dictionary in ascending order  {'apple': 10, 'banana': 5, 'kiwi': 2, 'orange': 20}
Dictionary in descending order  {'orange': 20, 'kiwi': 2, 'banana': 5, 'apple': 10}
```

DATE: 15/10/2024

18. Merge two dictionaries.

PROGRAM

```
d1={"apple":10,"orange":20,"banana":5,"kiwi":2}
d2={"pineapple":50,"mango":30}
print(d1)
print(d2)
d1.update(d2)
print(d1)
print(d1|d2)
```

OUTPUT

```
{'apple': 10, 'orange': 20, 'banana': 5, 'kiwi': 2}
{'pineapple': 50, 'mango': 30}
{'apple': 10, 'orange': 20, 'banana': 5, 'kiwi': 2, 'pineapple': 50, 'mango': 30}
{'apple': 10, 'orange': 20, 'banana': 5, 'kiwi': 2, 'pineapple': 50, 'mango': 30}
```

DATE: 22/10/2024

19. Find gcd of 2 numbers.

PROGRAM

```
import math
x=int(input("Enter num1: "))
y=int(input("Enter num2: "))
print("Gcd : ",math.gcd(x,y))
```

OUTPUT

```
Enter num1: 10
Enter num2: 25
Gcd :  5

Enter num1: 18
Enter num2: 9
Gcd :  9
```

DATE: 22/10/2024

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

PROGRAM

```
ol=[int(i) for i in input("Enter the integers: ").split()]
print("List before: ",ol)
newlist=[i for i in ol if i%2!=0]
print("List after removing even numbers: ",newlist)
```

OUTPUT

```
Enter the integers 3 5 8 9 10 12 4 88
List before:  [3, 5, 8, 9, 10,12, 4, 88]
List after removing even numbers:  [3, 5, 9]

Enter the integers  7 8 11 4 77 16 5
List before:  [ 7, 8, 11, 4, 77, 16, 5]
List after removing even numbers:  [ 7, 11, 77, 5]
```