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:

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 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: 18/09/2023

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

Program

```
startyear=int(input("Enter the start year"))
endyear=int(input("Enter last year"))
print("List of leap years")
for year in range(startyear,endyear):
  if((year%4==0)and(year%100!=0)or(year%400==0)):
    print(year)
```

Output

```
Enter the start year2000

Enter last year2023

List of leap years
2000
2004
2008
2012
2016
2020
```

Date: 18/09/2023

3. list comprehensions.

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

Program

```
integer=[2,5,7,-5,-6,-9]
list=[i for i in integer if i>0]
print("Positive integers",list)
```

Output

```
Positive integers [2, 5, 7]
```

b. Square of N numbers

Program

```
list1=[1,2,3,4,5]
list2=[i**2 for i in list1]
print("Square list",list2)
```

Output

```
Square list [1, 4, 9, 16, 25]
```

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

Program

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

Output

```
Enter a word:elephant
['e', 'e', 'a']
```

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

Program

```
word=input("Enter a character=")

o=[ord(i) for i in word]
print(o)
```

Output

```
Enter a character=python
[112, 121, 116, 104, 111, 110]
```

Date: 18/09/2023

4. count the occurence of each word in a line of text.

Program

```
str=input("Enter a line of text: ")
str=str.split()
i=0
count=0
while i<len(str):
  count=0
  for j in str:
    if str[i]==j:
      count=count+1
  print(str[i],"present" ,count ,"times")
  i=i+1
```

Output

```
Enter a line of text: Artificial intelligence is the intelligence of machines or
software
Artificial present 1 times
intelligence present 2 times
is present 1 times
the present 1 times
intelligence present 2 times
of present 1 times
machines present 1 times
or present 1 times
```

Date: 18/09/2023

5. prompt the user for a list of integers.for all values greater than 100,store 'over' instead

Program

```
n=int(input("Enter the number of elements"))
list=[]
for i in range(n):
  a=int(input("enter the integer"))
  if a<100:
    list.append(a)
  else:
    list.append("over")
print(list)
```

Output

```
Enter the number of elements5
enter the integer2
enter the integer105
enter the integer3
enter the integer107
enter the integer7
[2, 'over', 3, 'over', 7]
```

Date: 20/09/2023

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

Program

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

Output

```
'a' occurs in Navya 2 times
'a' occurs in Niva 1 times
'a' occurs in Nila 1 times
```

Date: 20/09/2023

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

Program

```
list1=[1,3,5,7,9]
list2=[1,2,4,6,8,]
if(len(list1)==len(list2)):
  print("Same length")
else:
  print("Different length")

if(sum(list1)==sum(list2)):
  print("Sums are same")
else:
  print("Sums are different")

if(set(list1) & set(list2)):
  print("Common number present")
else:
  print("No common numbers")
```

Output

```
Same length
Sums are different
Common number present
```

Date: 20/09/2023

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

Program

```
word=str(input("Enter a word  "))

ch=word[0]
word=word.replace(ch,"$")
word=ch+word[1:]
print(word)
```

Output

```
Enter a word  pappaya
pa$$aya
```

Date: 20/09/2023

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

Program

```
word=str(input("Enter a word "))
new_word=word[-1]+word[1:-1]+word[0]
print(new_word)
```

Output

```
Enter a word universe
eniversu
```

Date: 27/09/2023

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

Program

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

Output

```
Enter the radius:3
Area of circle=  28.259999999999998
```

Date: 27/09/2023

11.Find biggest of 3 numbers entered.

Program

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

Output

```
Enter first numbers: 5
Enter second numbers: 7
Enter third numbers: 1
7 is largest number
Enter first numbers: 8
Enter second numbers: 3
Enter third numbers: 1
8 is largest number

Enter first numbers: 2
Enter second numbers: 5
Enter third numbers: 9
9 largest number
```

Date: 27/09/2023

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

Program

```
file=str(input("Enter a file name "))
ext=file.split(".")
print("File extension =",ext[-1])
```

Output

```
Enter a file name python.py
File extension = py
```

Date: 27/09/2023

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

Program

```
n=int(input("Enter the number of colours "))
list=[]
for i in range(n):
  colours=str(input("Enter the colours name  "))
  list.append(colours)
print(list)
print(list[0],list[-1])
```

Output

```
Enter the number of colours 5
Enter the colours name  black
Enter the colours name  violet
Enter the colours name  pink
Enter the colours name  rose
Enter the colours name  purple
['black', 'violet', 'pink', 'rose', 'purple']
black purple
```

Date: 4/10/2023

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

Program

```
n=int(input("Enter the number "))

c=n+(n*n)+(n*n*n)

print("Output:",c)
```

Output

```
Enter the  number 2

Output:14
```

Date: 4/10/2023

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

Program

```
clrlist1=set(["black","blue","pink","violet"])
clrlist2=set(["blue","purple","white"])
print(clrlist1-clrlist2)
```

Output

```
{'black', 'pink', 'violet'}
{'blue', 'pink', 'white'}
```

Date: 4/10/2023

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

Program

```
str1=str(input("Enter first string: "))

str2=str(input("Enter second string: "))

t1=str1[0]

t2=str2[0]

new_str=t2+str1[1:]+" "+t1+str2[1:]

print(new_str)
```

Output

```
Enter first string: hello
Enter second string: world
wello horld
```

Date: 4/10/2023

17.Sort dictionary in ascending and descending order.

Program

```
dic={2:"cake", 1:"shake", 5: "choclate", 3: "ice cream", 4: "snacks"}
dic2={}
for i in sorted(dic):
   dic2[i]=dic[i]
print(dic2)
dict3=dict(sorted(dic.items(),reverse=True))
print(dict3)
```

Output

```
{1: 'shake', 2: 'cake', 3: 'ice cream', 4: 'snacks', 5: 'choclate'}
{5: 'choclate', 4: 'snacks', 3: 'ice cream', 2: 'cake', 1: 'shake'}
```

Date: 9/10/2023

18.Merge two dictionaries.

Program

```
dict1={"fruit1":"mango","fruit2":"chikku"}
dict2={"fruit3":"strawberry","fruit4":"blueberry"}
print(dict1|dict2)
```

Output

```
{'fruit1': 'mango', 'fruit2': 'chikku', 'fruit3': 'strawberry', 'fruit4': 'blueberry'}
```

Date: 9/10/2023

19. Find gcd of 2 numbers.

Program

```
n1 = int(input("Enter number1:"))
n2 = int(input("Enter number2:"))
gcd = 1

for i in range(1, min(n1, n2)):
    if n1 % i == 0 and n2 % i == 0:
        gcd = i
print("GCD of", n1, "and", n2, "is", gcd)
```

Output

```
Enter number1:12
Enter number2:24
GCD of 12 and 24 is 6
```

Date: 9/10/2023

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

Program

```
list=[9,2,5,4,7]
for i in list:
  div=i%2
  if div==0:
    list.remove(i)
print(list)
```

Output

```
[9, 5, 7]
```

COURSE OUTCOME 2

Date: 9/10/2023

1.Program to find the factorial of a number.

Program

```
fact=1
a= int(input("Enter a number"))
for i in range (1,a+1):
  fact=fact*i
print("Factorial of the number is",fact)
```

Output

```
Enter a number6
Factorial of the number is 720
```

Date: 11/10/2023

2.Generate Fibonacci series of N terms

Program

```
n=int(input("Enter the limit: "))
n1, n2 = 0, 1
print("Fibonacci Series:", n1, n2, end=" ")
for i in range(2, n):
n3 = n1 + n2
 n1 = n2
 n2 = n3
 print(n3, end=" ")
print()
```

Output

```
Enter the limit: 5
Fibonacci Series: 0 1 1 2 3
```

Date: 11/10/2023

3.Find the sum of all items in a list

Program

```
sum=0
list=[5,50,100,70]
for elements in range(0,len(list)):
   sum = sum + list[elements]
print("Sum of the elements in the list are",sum)
```

Output

```
Sum of the elements in the list are 225
```

Date: 11/10/2023

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

```
result=[]
for i in range(200,1000):
 if all(int(x)%2==0 for x in str(i)):
  if int(i**0.5)**2==i:
    result.append(i)
print("List of numbers : ",result)
```

Output

```
List of numbers :  [400, 484]
```

Date: 11/10/2023

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 the number of rows"))
for i in range(1,n+1):
  for j in range(1,i+1):
    print (j*i, end=" ")
  print()
```

Output

```
Enter the number of rows4
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

```
string=input("Enter the String: ")

for i in string:
    frequency = string.count(i)
    print(str(i) + ": " + str(frequency), end=", ")
```

Output

```
Enter the String: how are you
h: 1, o: 2, w: 1,  : 2, a: 1, r: 1, e: 1,  : 2, y: 1, o: 2, u: 1,
```

Date: 16/01/2023

Program

7.Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’

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

Output

```
Enter a string:play
playing
```

Date: 16/10/2023

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

Program

```
list=["one","two","three" ]
longest=max(list,key=len)
print(longest)
```

Output

```
three
```

Date: 16/10/2023

9.Construct following pattern using nested loop

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

Program

```
n=int(input("Enter number of rows:"))
for i in range(1,n+1):
  print("*"*i)
for i in range(n-1,0,-1):
  print("*"*i)
```

Output

```
Enter number of rows:5
*
**
***
****
*****
****
***
**
*
```

Date: 16/10/2023

10.Generate all factors of a number.

Program

```
num=int(input("Enter a number"))
factor=0
print('factors')
for i in range(1,num+1):
  if num%i==0:
   print(i)
```

Output

```
Enter a number75
factors
1
3
5
15
25
75
```

Date: 16/10/2023

11.Write lambda functions to find area of square, rectangle and triangle.

Program

```
a=int(input("Enter the side of the square"))
square= lambda a :a*a
print("Area of square is:",square(a))
b=int(input("Enter the breadth of the rectangle"))
l=int(input("Enter the length of the rectangle"))
rectangle = lambda b,l : b*l
print("Area of rectangle is:",rectangle(b,l))
b=int(input("Enter the base of the triangle"))
h=int(input("Enter the height of the triangle"))
triangle = lambda b,h : b*h*0.5
print("Area of triangle is:",triangle(b,h))
```

Output

```
Enter the side of the square2
Area of square is: 4
Enter the breadth of the rectangle4
Enter the length of the rectangle3
Area of rectangle is: 12
Enter the base of the triangle4
Enter the height of the triangle2
Area of triangle is: 4.0
```