**CO1 Program**

**Program 1**

Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE

**IDE(Integrated development environment)**

An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger.

Eg:- NetBeans, Eclipse, IntelliJ, and Visual Studio

**Text Editor**

A text editor is a type of computer program that edits plain text. Such programs are sometimes known as "notepad" software, following the naming of Microsoft Notepad.

Eg:- Atom, Notepad, Notepad++

**Code analysis tool**

Code analysis tool is a method of debugging by examining source code before a program is run. It's done by analyzing a set of code against a set (or multiple sets) of coding rules. 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.

Eg:- PVS-Studio, Raxis, CodeSonar

**PyCharm**

**Pycharm** is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

**Spyder**

**Spyder** is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. ... Spyder uses Qt for its GUI and is designed to use either of the PyQt or PySide Python bindings.

**RODEO**

**RODEO**, that is an opensource python IDE and has been brought up by the folks at yhat, is a development environment that is lightweight, intuitive and yet customizable to its very core and also contains all the features mentioned above that were searched for so long

**Thonny**

**Thonny** is an integrated development environment for Python that is designed for beginners. It supports different ways of stepping through the code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap.

**Atom**

**Atom** is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in JavaScript, and embedded Git Control. ... Its developers call it a "hackable text editor for the 21st Century", as it is fully customizable in HTML, CSS, and JavaScript.

**Program 2**

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

a=int(input("enter the starting year="))

b=int(input("enter the end year="))

if(a<b):

print("Leap year")

for i in range(a,b):

if(i%4==0 and i%100!=0):

print(i,end=" ")

**Output**

enter the starting year=2021

enter the end year=2050

Leap year

2024 2028 2032 2036 2040 2044 2048

**Program 3(a)**

**List comprehensions:**

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

lt =[-10,20,35,-67,70]

r=[n for n in lt if n>=0]

print(r)

**output**

[20, 35, 70]

**program 3(b)**

**Square of N number**

n=int(input("Enter the limit"))

sqlist= [ i\*\*2 for i in range(1,n+1)]

print("Square of N numbers : ", sqlist)

**output**

Enter the limit4

Square of N numbers : [1, 4, 9, 16]

**program 3(c)**

**Form a list of vowels selected from a given word**

w =str(input("Enter the word :"))

print("The original string is : "+w)

print("The vowel are : ",end="")

for i in w:

if i in 'aeiouAEIOU':

print([i],end=" ")

**output**

Enter the word :Python Programming

The original string is : Python Programming

The vowel are : ['o'] ['o'] ['a'] ['i']

**program 3(d)**

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

word=input("Enter a word:")

print("Ordinal values corresponding to each element is:")

for i in word:

print(i,end=":")

print(ord(i),end=" ")

**output**

Enter a word:PYTHON

Ordinal values corresponding to each element is:

P:80 Y:89 T:84 H:72 O:79 N:78

**Program 4**

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

str1 = input("Enter a string : ")

word = str1.split()

count= []

for w in word:

count.append(word.count(w))

print("count of the occurrence:" + str(list(zip(word, count))))

**output**

Enter a string : Python programming

count of the occurrence:[('Python', 1)]

count of the occurrence:[('Python', 1), ('programming', 1)]

**Program 5**

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

n=[]

s=int(input("Enter a limit:"))

print("Enter {s} values")

for i in range(0,s):

n.append(int(input()))

print("\nThe list after assinging:\n")

for i in range(0,len(n)):

if(n[i]>=100):

print("over")

else:

print(n[i])

**output**

Enter a limit:2

Enter {s} values

24

199

The list after assinging:

24

Over

**Program 6**

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

list1=["a","d","c","a","a","a","a",]

c=list1.count("a")

print("Occurrence of a=",c)

**Output**

Occurrence of a= 5

**Program 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**

lst=[1,3,5,7,9,11,34]

lst1=[5,13,45,7,20,65,1]

s=int(0)

c=int(0)

if(len(lst)==len(lst1)):

print("Lists are of same length")

else:

print("Lists have different length")

for i in range(0,len(lst) and len(lst1)):

s=s+lst[i]

c=c+lst1[i]

if(s==c):

print("equal sum")

else:

print("not same sum")

print("Elements that matched are:")

l=[]

for i in range(0,len(lst)):

for j in range(0,len(lst1)):

if lst[i]==lst1[j]:

l.append(lst[i] and lst1[j])

else:

continue

print(l)

**output**

Lists are of same length

not same sum

Elements that matched are:

[1, 5, 7]

**Program 8**

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

str2="onion"

char=str2[0]

str2=str2.replace(char,'$')

str2=char+str2[1:]

print(str2)

**output**

oni$n

**Program 9**

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

str=input("Enter a string:")

nstr=str[-1:] +str[1:-1] + str[:1]

print("New string : ",nstr)

**Output**

Enter a string:Python

New string : nythoP

**Program 10**

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

r=float(input("Enter the radius="))

pi=3.14

area=pi\*r\*r

print("Area of circle=",area)

**output**

Enter the radius=5

Area of circle= 78.5

**Program 11**

**Find biggest of 3 numbers entered**

x=int(input("enter the number="))

y=int(input("enter the number="))

z=int(input("enter the number="))

if(x>y):

if(x>z):

print("x is large",x)

else:

print("z is large",z)

else:

if(y>z):

print("y is large",y)

else:

print("z is large",z)

**output**

enter the number=5

enter the number=7

enter the number=2

y is large 7

**Program 12**

**Accept a file name from user and print extension of that**

x,y=input("Enter the file name=").split(".")

print("Extension of file=",y)

**Output**

Enter the file name=program.java

Extension of file= java

**Program 13**

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

a=[]

for i in range(3):

b=input("enter the color:")

a.append(b)

print(a)

print(a[0])

print(a[2])

**output**

enter the color:red

enter the color:blue

enter the color:green

['red', 'blue', 'green']

red

green

**Program 14**

**Accept an integer n and compute n+nn+nnn**

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

x=int(n+n+n)

y=int(n+n)

z=int(n)

print("n+nn+nnn=",x+y+z)

**Output**

Enter the number n=5

n+nn+nnn= 615

**Program 15**

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

list1=set(["Red","blue","green","white"])

list2=set(["Yellow","blue","Black","white"])

c=list1.difference(list2)

print(c)

**Output**

{'green', 'Red'}

**Program 16**

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

str1="python"

str2="java"

s1=str1[0]

s2=str2[0]

print(s2+str1[1:]+" "+s1+str2[1:])

**Output**

jython pava

**Program 17**

**Sort dictionary in ascending and descending order.**

import operator

d={'apple':20,'orange':10,'mango':12}

print("orginal dictionary")

print(d)

print("ascending")

sdk=sorted(d.items(),key=operator.itemgetter(0))

print(sdk)

print("descending")

sdk=sorted(d.items(),key=operator.itemgetter(0),reverse=True)

print(sdk)

**Output**

orginal dictionary

{'apple': 20, 'orange': 10, 'mango': 12}

ascending

[('apple', 20), ('mango', 12), ('orange', 10)]

descending

[('orange', 10), ('mango', 12), ('apple', 20)]

**Program 18**

**Merge two dictionaries**

dict1={"name":'Aswin',"age":18}

dict2={"class":'mca',"year":2021}

print("Dictionary 1=",dict1)

print("Dictionary 2=",dict2)

d=dict1.copy()

d.update(dict2)

print("Merged dictionary",d)

**Output**

Dictionary 1= {'name': 'Aswin', 'age': 18}

Dictionary 2= {'class': 'mca', 'year': 2021}

Merged dictionary {'name': 'Aswin', 'age': 18, 'class': 'mca', 'year': 2021}

**Program 19**

**Find gcd of 2 numbers.**

a=int(input("Enter the 1st number="))

b=int(input("Enter the 2nd number="))

i=1

gcd=0

while(i<=a and i<=b):

if(a%i==0 and b%i==0):

gcd=i

i=i+1

print("GCD =",gcd)

**Output**

Enter the 1st number=120

Enter the 2nd number=5

GCD= 5

**Program 20**

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

num = [7,8, 120, 25, 44, 20, 27]

print( "Original list:",num)

num = [x for x in num if x%2!=0]

print("list after removing Even numbers:",num)

**output**

Original list: [7, 8, 120, 25, 44, 20, 27]

list after removing Even numbers: [7, 25, 27]