1. **SIMPLE INTEREST**

**Program coding:**

p=int(input("enter principal amount: "))

n=int(input("enter the number of years: "))

r=float(input("enter the rate of interest: "))

SI=p\*n\*r/100

print("the SI is ",SI)

**OUTPUT:**

enter principal amount: 3000

enter the number of years: 3

enter the rate of interest: 2

the SI is 180.0

1. **INCOME TAX CALCULATION**

**PROGRAM CODING:**

#nested if to find incometax

income=int(input("Enter the income :"))

if(income<200000):

print("No income tax")

elif(income>200000 and income<400000):

tax=(income-200000)\*0.1

print(tax)

elif(income>400000 and income<600000):

tax=(20000+income-200000)\*0.2

print(tax)

elif(income>600000):

tax=(20000+40000+income-600000)\*0.3

print(tax)

**OUTPUT:**

Enter the income :3000000

738000.0

1. **MULTIPLICATION TABLE USING FOR LOOP**

**PROGRAM CODING:**

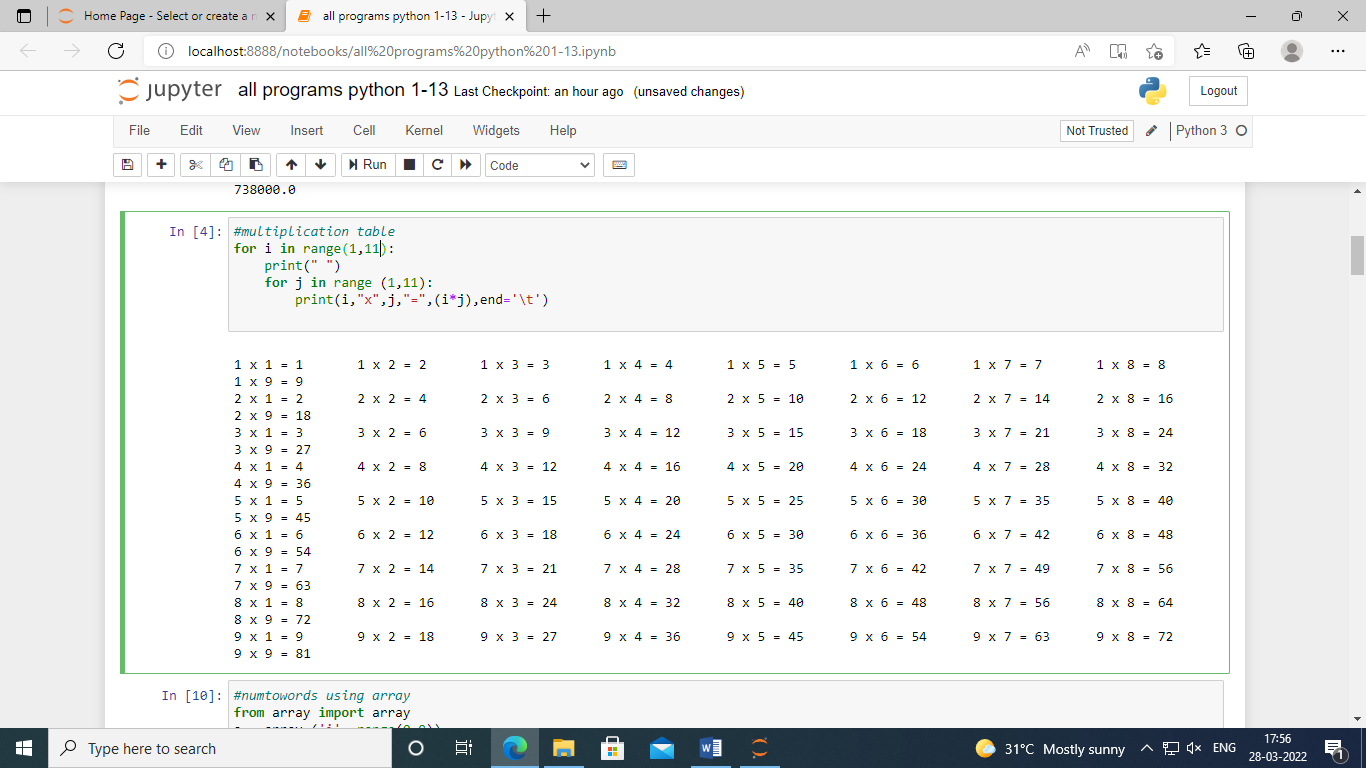
for i in range(1,11):

print(" ")

for j in range (1,11):

print(i,"x",j,"=",(i\*j),end='\t')

**OUTPUT:**



1. **NUMBER TO WORDS USING ARRAY**

**PROGRAM CODING:**

from array import array

a=array('i',range(3))

x=int(input("Enter a 3 digit number:"))

a[0]=int(x/100)

a[1]=int(x/10)%10

a[2]=int(x%10)

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

if a[i]==0:

print("zero")

elif a[i]==1:

print("one")

elif a[i]==2:

print("two")

elif a[i]==3:

print("three")

elif a[i]==4:

print("four")

elif a[i]==5:

print("five")

elif a[i]==6:

print("six")

elif a[i]==7:

print("seven")

elif a[i]==8:

print("eight")

else:

print("nine")

**OUTPUT:**

enter a 3 digit number200

two zero zero

1. **NUMBER TO WORDS USING FUNCTION**

**PROGRAM CODING:**

def text\_eq(x):

if x==0:

y="zero"

elif x==1:

y="one"

elif x==2:

y="two"

elif x==3:

y="three"

elif x==4:

y="four"

elif x==5:

y="five"

elif x==6:

y="six"

elif x==7:

y="seven"

elif x==8:

y="eight"

else:

y="nine"

return y

i=int(input("Enter a 3 digit num: "))

d1=int(i%10)

print(text\_eq(d1))

d2=int(i/10%10)

print(text\_eq(d2))

d3=int(i/100)

print(text\_eq(d3))

**OUTPUT:**

Enter a 3 digit num: 100

one zero zero

1. **STUDENT INFORMATION USING LIST**

**PROGRAM CODING:**

a=[]

n=int(input("enter total number of students: "))

for i in range(0,n):

dno=input("enter the dept number: ")

a.append(dno)

name=input("enter the name: ")

a.append(name)

dob=input("enter the dob: ")

a.append(dob)

fname=input("enter the fname: ")

a.append(fname)

score=input("enter the score: ")

a.append(score)

print(a)

**OUTPUT:**

enter total number of students: 1

enter the dept number: 19-uca-009

enter the name: bovas

enter the dob: 30-01-2002

enter the fname: chitta

enter the score: 89

['19-uca-009', 'bovas', '30-01-2002', 'chitta', '89']

1. **STUDENT DETAILS USING DICTIONARY**

**PROGRAM CODING:**

dno={1:"19-UCA-001",2:"19-UCA-002",3:"19-UCA-055"}

name={1:"Kewin",2:"Daniel",3:"Bovas"}

dob={1:"06-04-2002",2:"12-06-2000",3:"25-05-2001"}

mobile={1:"8056045990",2:"7305487090",3:"9234568740"}

for key in dno:

print(key, dno[key],name[key],dob[key],mobile[key])

**OUTPUT:**

1 19-UCA-001 Kewin 06-04-2002 8056045990

2 19-UCA-002 Daniel 12-06-2000 7305487090

3 19-UCA-055 Bovas 25-05-2001 9234568740

1. **STANDARD DEVIATION**

**PROGRAM CODING:**

import math

n = int(input("enter the number of values:"))

weight =[]

s = 0

for i in range (n):

w = int(input("enter the weight:"))

weight.append(w)

s = s+w

print(weight)

average = int(s/n)

d = 0

for i in range (n):

d = (weight[i] - average)\*(weight[i] - average)

sd = math.sqrt(d)

print("The standard deviation is:",sd)

**OUTPUT:**

Enter the number of Values: 2

Enter the weight30

Enter the weight30

[30, 30]

The standard deviation is : 42.42640687119285

1. **STACK IMPLEMENTATION MODULES**

**PROGRAM CODING:**

stack = []

while(True):

x = int(input("Enter your choice: [1]-Insert [2]-Delete [3]-Display"))

if(x==1):

v = int(input("Enter any integer value: "))

stack.append(v)

print(stack)

elif(x==3):

print(stack)

elif x ==2 :

stack.pop()

print(stack)

**OUTPUT:**

Enter your choice: [1]-Insert [2]-Delete [3]-Display1

Enter any integer value: 23

[23]

Enter your choice: [1]-Insert [2]-Delete [3]-Display1

Enter any integer value: 24

[23, 24]

Enter your choice: [1]-Insert [2]-Delete [3]-Display2

[23]

Enter your choice: [1]-Insert [2]-Delete [3]-Display3

[23]

1. **AGE CALCULATION – OOPS**

**PROGRAM CODING:**

def ageCalculator(y, m, d):

import datetime

today = datetime.datetime.now().date()

dob = datetime.date(y, m, d)

age = int((today-dob).days / 365.25)

print(age ,"YEARS" )

ageCalculator(1998, 9, 3)

**OUTPUT:**

23 YEARS

**11.STUDENT PROFESSOR TYPE OF INHERITANCE**

**PROGRAM CODING:**

#single inheritance

class Parent:

def myfun\_p(self):

print("I'm the Parent")

class Child(Parent):

def myfun\_c(self):

print("I'm the Child")

obj = Child()

obj.myfun\_p()

obj.myfun\_c()

**OUTPUT:**

I’m the Principal

I’m a Teacher

I’m a student

#multiple inheritance

class Name:

name = ""

def myfun1(self):

print(self.name)

class Surname:

surname = ""

def myfun2(self):

print(self.surname)

# Multi Child class

class Student(Name, Surname):

def parents(self):

print("Name :", self.name)

print("Surname :", self.surname)

stud = Student()

stud.name = "bovas"

stud.surname = "chitta"

stud.parents()

**OUTPUT:**

Name : bovas

Surname : chitta

#multi-level inheritance

class Principal:

def principal(self):

print("I’m the Principal")

class Teacher(Principal):

def teacher(self):

print("I’m a Teacher")

class Student(Teacher):

def student(self):

print("I’m a student")

d = Student()

d.principal()

d.teacher()

d.student()

**OUTPUT:**

I’m the Principal

I’m a Teacher

I’m a student

#hierarchial inheritance

# Python program to demonstrate

# Hierarchical inheritance

# Base class

class HOD:

def func1(self):

print("I'm the HOD")

# Derived class1

class CO\_ORDINATOR(HOD):

def func2(self):

print("I'm the co-ordinator under HOD")

# Derivied class2

class staff(HOD):

def func3(self):

print("I'm a STAFF under HOD")

# Driver's code

o1 = CO\_ORDINATOR()

o2 = staff()

o1.func1()

o1.func2()

o2.func3()

**OUTPUT:**

I'm the HOD

I'm the co-ordinator under HOD

I'm a STAFF under HOD

# Python program to demonstrate

# hybrid inheritance

class College:

def func1(self):

print("Loyola College")

class Student1(College):

def func2(self):

print("This function is in student 1. ")

class Student2(College):

def func3(self):

print("This function is in student 2.")

class Student3(Student1, College):

def func4(self):

print("This function is in student 3.")

# Driver's code

o = Student3()

o.func1()

o.func2()

**OUTPUT:**

Loyola College

This function is in student 1.

1. **CART UPADTING**

**PROGRAM CODING:**

class Order:

def \_\_init\_\_(self, cart, customer):

self.cart = list(cart)

self.customer = customer

def \_\_add\_\_(self, other):

new\_cart = self.cart.copy()

new\_cart.append(other)

return Order(new\_cart, self.customer)

def \_\_radd\_\_(self, other):

new\_cart = self.cart.copy()

new\_cart.insert(0, other)

return Order(new\_cart, self.customer)

order = Order(['pant', 'towel'], 'flipkart')

order = order + 'belt'

order.cart

order = 'tie' + order

order.cart

**OUTPUT:**

**['tie', 'pant', 'towel', 'belt']**

1. **SHAPE SQUARE OVERRIDING**

**PROGRAM CODING:­­­­­­­­­­**

# Parent class

class Shape:

# properties

data1 = "abc"

# function no\_of\_sides

def no\_of\_sides(self):

print("My sides need to be defined. I am from shape class.")

# function two\_dimensional

def two\_dimensional(self):

print("I am a 2D object. I am from shape class")

#Sub-class

class Square(Shape):

data2 = "xyz"

def no\_of\_sides(self):

print("I have 4 sides. I am from Square class")

def color(self):

print("I have teal color. I am from Square class.")

# Create an object of Square class

sq = Square()

# Override the no\_of\_sides of parent class

sq.no\_of\_sides()

# Will inherit this method from the parent class

sq.two\_dimensional()

# It's own method - color

sq.color()

**OUTPUT:**

I have 4 sides. I am from Square class

I am a 2D object. I am from shape class

I have teal color. I am from Square class.