**Day 1**

**—------------------------------**

**# -\*- coding: utf-8 -\*-**

**"""**

**Created on Thu Mar 23 10:32:00 2023**

**@author: Maanav**

**"""**

**# Python**

**# High level programming**

**# write computer instructions- easy to understand**

**# Guido Van Rossam - 1989**

**# 1991-public**

**# First prog language for beginners**

**# #include<stdio.h>**

**# void main()**

**# {**

**#  initili**

**#  printf("hello world");**

**#  }**

**#print("hello world")**

**# class add{**

**#         PSVM(S a[])**

**#         {**

**#             int a,b,c;**

**#             a=10;**

**#             b=20;**

**#             c=a+b;**

**#             }**

**#         }**

**# a=10**

**# b=20**

**# print("the sum is",a+b)**

**#functionoal program features+OOP+scripting +modular prog = PYTHON**

**# Where we can use python**

**# ------------**

**# -Data analysis**

**# -ML AI**

**# -web application**

**# -desktop**

**# -developing games**

**# -IOT**

**# ...**

**# ...**

**# Why Python**

**# ---------**

**# -Easy to learn and simple - 30+ keywords**

**# -less code**

**# -Library- numpy,pandas**

**# -OOPS + procedure**

**# -reduce resources**

**# -platform independant- JVM |PVM is convert into machine understandble form**

**# -OPEN SOURCE | FREEWARE(without license)**

**# -source code is open ->customize**

**# -Jython -work with java appln**

**# -Cpython,Ironpython, Ananconda python**

**# -Dynamically typed - not required to declare type for variables**

**# -C,java - statically typed**

**#interpred - python intepreter**

**#extensible- we can use other langugage programs in python**

**# a=3+2j**

**# b=20**

**# # print(a+b)**

**# print(type(a))**

**# import keyword**

**# print(keyword.kwlist)**

**#identifiers**

**#Name in python**

**#case sensitive**

**# a=10**

**# class or function or module variable name:**

**# starts with aplahbets ,digits 0 to 0 , \_**

**# abc\_name=12345**

**# \_add\_**

**#\_ private     \_\_ strongly private**

**# total=100**

**# TOTAL=2000**

**# print(id(total))**

**# print(id(TOTAL))**

**# true**

**# True**

**#Datatype**

**# int**

**# float**

**# complex()**

**# bool()**

**# string**

**# list**

**# tuple()**

**# set**

**# dict()**

**# bytes()**

**# type()- checl variable type**

**# id() -address of object**

**# print()**

**# Integer type**

**# ----------------**

**# Decimal - 0 to 9**

**# Binary - 0 & 1 . starts with 0b or 0B**

**# Octa - 0 to 7 or 0o or 0O**

**# Hexa - 0 to 9, a-f or 0x or 0X**

**# a=0XBeef**

**# print(type(a))**

**#In-Built functions for base conversions**

**#bin- base to binary**

**# print(bin(15))**

**#oct**

**#print(oct(10))**

**#hex**

**# f=1.234**

**# print(type(f))**

**# Exponential values- represent big values in less memory**

**# f1=1.2e3**

**# print(f1)**

**# a=0B111+3j**

**# perform - add,sub**

**# b=True**

**# print(type(b))**

**# a=10**

**# d=20**

**# print(a<d)**

**#print(True+True)**

**#inbuilt functions for type casting**

**# int()**

**# float()**

**# complex()**

**# bool()**

**# str()**

**#print(int(123.456))**

**#print(int(10+3j))**

**#print(int("10.5"))**

**#print(int("ob1111"))**

**#all fundamental data types are IMMUTABLE.**

**# a=10**

**# b=10**

**# print(a is b)**

**# print(id(a))**

**# print(id(b))**

**#bytes data types - group of byte numbers just like an array**

**# x=[10,20,30,40]**

**# b=bytes(x)**

**# print(type(b))**

**# print(b[0])**

**#bytearray - objects can be modified**

**######### try it yourself##############**

**# operators**

**# -----------**

**# arith**

**# relational**

**# logical**

**# bitwise**

**# assignment**

**# special**

**#special operators**

**# 1.Identity operator (is , is not)- address comparision**

**# 2.Memebership operator(in, not in)- check whether the given object present in the given collection or not**

**# ad1 ad2->points to same object =>True**

**# if not returns false:**

**# a=10**

**# b=10**

**# print(a is b)**

**# x=True**

**# y=False**

**# print(x is y)**

**# is->address comparision**

**# == ->content content**

**# x="hello all learning python is very simple"**

**# print('h' in x)**

**# print('z' in x)**

**#dynamic input from the keyboard**

**#input**

**# x=int(input("enter value"))**

**# y=int(input("enter value2"))**

**# print(type(x))**

**# print("the sum is",(x+y))**

**# x=input("enter value")**

**# y=input("enter value2")**

**# print(type(x))**

**# print("the sum is",(x+y))**

**#read multiple values from the keyboard in a single line**

**# a,b=[int(x) for x in input("enter 2 numbers").split()]**

**# print("product is",a\*b)**

**#WAP to read multiple float numbers from the keyboard with '-' as**

**#a seperator and print their product**

**#eval -> take string and evaluate the result**

**# x=eval("1+2+3")**

**# print(x)**

**#ouput statement => print()**

**#formatted string**

**# #%d, %f %c**

**# %i =>int**

**# %d =>int**

**# %f =>float**

**# %s =>string**

**# a=10**

**# b=20**

**# c=30**

**# print("a value is %i" %a)**

**# print("b value is %d and c value is %d" %(b,c))**

**#replacement operator {} ##################################**

**# name='python'**

**# salary=3000**

**# age=35**

**# print("hello {0} your salary is {1} and your age is {2}".format(name,salary,age))**

**# print("hello {x} your salary is {y} and your age is {z}".format(x=name,y=salary,z=age))**

**#Flow Control**

**# conditional statement - if,if elif, if elif else**

**# Transfer statements - break, continue , pass**

**# Iterative statement - for ,while**

**# if condition:**

**#     action**

**# else:**

**#     action**

**# name= input("enter name")**

**# if name=='python':**

**#     print("hello python good afternoon")**

**# else:**

**#     print("hello guest good afternoon")**

**# print("Welcome")**

**# synatx if-elif-else**

**# if condn1:**

**#     action**

**# elif condn2:**

**#     action**

**# elif condn3:**

**#     action**

**# else:**

**#     default action**

**#iterative statements**

**#for**

**#while**

**#for x in sequence:**

**# body**

**#sequence : list,tuple,range,string**

**# s="python programming"**

**# for x in s:**

**#     print(x)**

**# Print the characters and its index value**

**# sample output: the character at 0 index is : P**

**# for x in range(10):**

**#     print("hello")**

**# for x in range(11):**

**#     print(x)**

**# for x in range(21):**

**#     if (x%2!=0):**

**#         print(x)**

**# for x in range(10,0,-1):**

**#     print(x)**

Task 1:

Imagine that you are in a desert and all of a sudden, a space shuttle lands in front of you. An alien walks out of the space shuttle and greets you. Write a Python program to welcome this friendly alien to our planet - Earth.  Get the name of the alien from the user and display the welcome message as given in the sample output.

Sample Input 1:

Enter the name:Naoto

Sample Output 1:

Hello Naoto! Welcome to our planet Earth.

Task 2:

 Write a Python program to calculate and display the income tax of an individual by giving his/her age and annual income. Refer to the Income Tax Slab given below and code accordingly.

Income Tax Slab:

Age <=60 and income<= 2, 50,000/ : 0% tax

Age <=60  and income>2,50,000  and income<=  5,00,000 : 10% tax

Age <=60  and income>5,00,000  and income<=  10,00,000   : 20% tax

Age <=60  and income above 10, 00,000/-     : 30% tax

Age >60 and age<=80  , income<= 3, 00,000/ : 0% tax

Age >60 and age<=80 ,  income > 3,00,000 and income <=5,00,000 :10% tax

Age >60 and age<=80 ,  income > 5,00,000 and income <=10,00,000 : 20% tax

Age >60 and age<=80 ,  income>10,00,000 : 30% tax

Age >80 and income<= 5, 00,000/ : 0% tax

Age >80 ,  income > 5,00,000 and income <=10,00,000 : 20% tax

Age >80  and age<=100,  income >10,00,000 : 30% tax.

Note:

● If the age entered by the user is below 18 and above 100, then, display  the message "Invalid Age" and stop the program.

● If the income entered by the user is a negative number, then, display the message " Invalid Income " and stop the program.

Sample Input 1:

Enter the age:

50

Enter the income:

250000

Sample Output 1:

The Tax amount is: 0.00

Sample Input 2:

Enter the age:

40

Enter the income:

500000

Sample Output 2:

The Tax amount is: 50000.00

Sample Input 3:

Enter the age:

17

Sample Output 3:

Invalid Age

Sample Input 4:

Enter the age:

40

Enter the income:

-500000

Sample Output 4:

Invalid Income

age = int(input("Enter your age: "))

income = int(input("Enter your annual income: "))

if age <= 60:

if income <= 250000:

tax = 0

elif income <= 500000:

tax = 0.1 \* (income)

elif income <= 1000000:

tax = 0.2 \* (income)

else:

tax = 0.3 \* (income)

elif age <= 80:

if income <= 300000:

tax = 0

elif income <= 500000:

tax = 0.1 \* (income)

elif income <= 1000000:

tax = 0.2 \* (income)

else:

tax = 0.3 \* (income)

else:

if income <= 500000:

tax = 0

elif income <= 1000000:

tax = 0.2 \* (income)

else:

tax = 0.3 \* (income)

print("Your income tax is:", tax)

Task 3:

# Initializing an empty list to store the student details

students = []

# Getting the number of students to be added

num\_of\_students = int(input("Enter the number of students: "))

# Checking if the number of students is greater than zero

if num\_of\_students <= 0:

print("Invalid Input")

exit()

# Loop to get the details of each student

for i in range(num\_of\_students):

print(f"\nEnter details of student {i+1}:")

name = input("Name: ")

age = int(input("Age: "))

location = input("Training location: ")

# Checking if the age of the student is valid

if age <= 10 or age >= 20:

print("Invalid Input")

exit()

# Adding the student details to the list of dictionaries

students.append({'Name': name, 'Age': age, 'Location': location})

# Getting the training location to search for students

search\_location = input("Enter the training location to find students: ")

# Checking if the entered training location is valid

if search\_location not in [student['Location'] for student in students]:

print("Invalid location")

exit()

# Printing the names of the students from the entered training location

print(f"\nStudents from {search\_location}:")

for student in students:

if student['Location'] == search\_location:

print(student['Name'])

Task 4 :

# Getting the number of plots

n = int(input("Enter the number of plots: "))

# Checking if the number of plots is valid

if n <= 0 or n > 20:

print("Invalid Input")

exit()

# Getting the plot numbers

plots = []

for i in range(n):

plot = float(input(f"Enter the plot number {i+1}: "))

if plot <= 0:

print("Invalid Input")

exit()

plots.append(plot)

# Calculating the sum of odd and even plot numbers

even\_sum = sum([plot for plot in plots if plot % 2 == 0])

odd\_sum = sum([plot for plot in plots if plot % 2 == 1])

# Calculating the average and displaying with 2 decimal places

average = round((even\_sum + odd\_sum) / 2, 2)

print(f"The password for the file is: {average}")