Module-6: Python Fundamentals

Lab exercise

* 1.INTRODUCTION TO PYTHON:

🡪Que-1: Write a Python program that prints "Hello, World!".

Ans

Print ("hello welcome to python")

🡪Que-2: Set up Python on your local machine and write a program to display your name.

Ans:

Step-1: Install python

* Install latest version of python from original website.

Step 2: Choose an Editor

You can use:

* **IDLE** (comes with Python)
* **VS Code** (lightweight and powerful)
* **PyCharm** (IDE for large projects)
* **Jupyter Notebook** (for data science and notes)

Step 3: save program with .py extension ex.first.py

Step 4:print my name:

Example:

print("Good Morning, My Name is Yogesh Patel")

-------------------------------------------------------------------------------------

* 2. Programming Style:

🡪Write a Python program that demonstrates the correct use of indentation, comments, and variablesfollowing PEP 8 guideline.

name="Yogesh"

print(name)

-------------------------------------------------------------------------------------

* 3:Core Python Concepts

# i):-Write a Python program to demonstrate the creation of variables and different data types.

roll=101

name='yogesh'

percentage=89.98

fees=1000000

print(type(roll))

print(type(name))

print(type(percentage))

print(type(fees))

# ii):- Practical Example 1: How does the Python code structure work?

"""

Understanding Python Code Structure

Python has a very simple, readable, and indentation-based structure.

Unlike C, C++, or Java — there are no curly braces {} to mark blocks.

Instead, indentation (spaces or tabs) decides the code structure.

Basic Components of Python Structure

Comments – start with # (for documentation or notes).

# This is a comment

Docstrings – multi-line documentation using triple quotes.

#This program demonstrates Python structure.

Import Statements – to use external or built-in modules.

import math

from datetime import datetime

Function & Class Definitions – reusable blocks of code.

def greet(name):

    print(f"Hello, {name}")

class Person:

    def \_\_init\_\_(self, name):

        self.name = name

Main Code Block – executed when the file runs directly.

if \_\_name\_\_ == "\_\_main\_\_":

    greet("Yogesh")"""

# iii):- Practical Example 2: How to create variables in Python?

"""

Variable: A name which can  store a value of different data type

Rules For Variable Declaration:

1)we can not use any keyword or reserved word as variable name

  ex. int,float

2)we can use a-z,A-z ,underscore and 0-9 in variable name declaration.

ex. name1,number,sum

3)we can not use white in variable name also use camel case or underscore.

ex. first name  X

    firstName  # lower camel case valid

    FirstName  # Uppercamel case valid

    First\_name # underscore valid

4) we can't use digit as first letter of variable.

ex 1num   X

   2num   X

   num1 #valid

   num2 # valid

5)Python is case sensitive language so AGE and age Are different

name='yogesh'   # string variable

num=10          # int  variable

percentage=89.98 # float variable

flag=true   # boolean variable

we can also declare multiple  varible

x,y,z=10,20,30

We can also assign same value to differnt variable using assignment operator

a=b=c='Python'

"""

# iv):- Practical Example 3: How to take user input using the input() function.

"""

input(): input is function that can accept user input we can use it to take dynamic value from user

syntax

input("Prompt message")

by default input function convert input data into string

so,when we want to  use some calculation we are need to type convertion.

Type conversion: Type casting use to convert one type data into another type data

syntax

variable\_name=int(input("prompt message"))

"""

name=input("Enter your name= ")

R\_num=int(input("Enter Your Roll Number= "))

print(name)

print(R\_num)

# v):- Practical Example 4: How to check the type of a variable dynamically using type()

"""

Type(): By using type function we can check what type of data variable can hold or store

syntax

type(varible\_name)

"""

num=10

print(type(num))

* 4.Conditional Statement:
* #Que- 4 Conditional statement
* """
* #Practical Example 5: Write a Python program to find greater and
* #  less than a number using if\_else.
* """
* num1=int(input("Enter number 1= "))
* num2=int(input("Enter Number 2= "))
* if num1>num2:
* print("Number 1 is Greter than number two")
* else:
* print("Number 1 is less than Number two ")
* #----------------------------------------------------------------------------------------
* """
* • Practical Example 6: Write a Python program to check if a number is prime using if\_else.
* """
* def prime(num,divisor):
* if num<=1:
* return 0
* elif num==divisor:
* return 1
* elif num%divisor==0:
* return 0
* return prime(num,divisor+1)
* num=int(input("Enter Number= "))
* if prime(num,2):
* print(f"{num} is prime number")
* else:
* print(f"{num} is not prime number")
* #----------------------------------------------------------------------------------------
* """• Practical Example 7: Write a Python program to calculate grades based on percentage using
* if-else ladder. """
* percentage=int(input("Enter Your Percentage= "))
* if percentage>=90:
* print("Grade:A+")
* elif percentage>=75:
* print("Grade:A")
* elif percentage>=55:
* print("Grade:B")
* elif percentage>=35:
* print("Grade:C")
* else:
* print("Grade:D")
* #-----------------------------------------------------------------------------------------
* """• Practical Example 8: Write a Python program to check if a person is eligible to donate blood
* using a nested if"""
* age=int(input("Enter Your Age= "))
* weight=int(input("Enter Your Weight= "))
* if age>=18:
* if weight>=50:
* print("You are eligible for Blood Donation")
* else:
* print("You are Not Eligible for Blood Donation Because Your weight is less than 50kg")
* else:
* print("You are Not Eligible for Blood Donation Because Your age is less than 18 year")
* #-----------------------------------------------------------------------------------------

----------------------------------------------------------------------------------

* 5.Looping(For,While):-

🡪"""

Que-5.Loops (For,While)

"""

#Practical Example 1: Write a Python program to print each fruit in a list using a simple for

#loop. List1 = ['apple', 'banana', 'mango']

list1=['apple','banana','mango']

for i in list1:

    print(i)

"""-------------------------------------------------------------------------------------------------------"""

#• Practical Example 2: Write a Python program to find the length of each string in List1.

list1=['apple','banana','mango']

for i in list1:

    print(f"{i} {len(i)}")

"""--------------------------------------------------------------------------------------------------------"""

#• Practical Example 3: Write a Python program to find a specific string in the list using a simple

#for loop and if condition.

list1=['apple','banana','mango']

search\_str=input("Enter Searching String Name= ")

if search\_str in list1:

    print("String Found")

else:

    print("String not found")

"""--------------------------------------------------------------------------------------------------------"""

"""• Practical Example 4: Print this pattern using nested for loop:

markdown

Copy code

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

"""

row=int(input("Enter Row = "))

for i in range(1,row+1):

    for j in range(1,i+1):

        print("\*",end="")

    print()

"""----------------------------------------------------------------------------------------------------------"""