**Python Datatypes Assignment (Up to Dictionaries)**

1. Add an integer and float. What is the result’s type?

a=6

b=8.2

result=a+b

Print(type(result))

Output: <class 'float'>

2.Create a string and access its:

* 1. First character
  2. Last character
  3. A substring from index 2 to 5

Program:

str1='Parveen'

print(str1[0])

print(str1[6])

print(str1[2:5])

Output:

P

N

rve

3.Concatenate two strings and print the result.

str1=Naina

str2=Parveen

print(str1+str2)

Output:

NainaParveen

4.Define list. What are the difference between List and Tuple.

List: A **list** is a mutable, ordered collection of items. This means you can modify a list

after its creation by adding, removing, or changing elements. Lists are defined using

square brackets [].

**Lists** are **mutable**, meaning you can modify them—add, remove, or change items.

**Tuples** are **immutable**; once created, their content and length can’t be changed

5.Write a programme to print a list in reverse order.

list1=[2,4,’Nainaa',6,10]

print(list1[::-1])

output: [10,6,’Nainaa’,4,2]

6.Create a tuple of 4 elements. Print the first and last element.

tup1=(4,6,8,10)

print(tup1[0],tup1[3])

Output:4 10

7.Try changing a value in a tuple. What happens?

Attempting to change a value in a tuple causes a **TypeError**, because tuples are

**immutable**—their contents can't be altered once created.

8.Create a dictionary of 3 students with their marks. Print the dictionary.

dict1={

'Naina':480,

' ‘Parveen':520,

'Aaliyaa':380

}

print(dict1)

output: {‘Naina’: 480, 'Parveen': 560, 'Aaliyaal': 380}

9.Access the marks of one student using their name.

dict1={

'Naina':480,

' ‘Parveen':520,

'Aaliyaa':380

}

print(dict1[‘Parveen’])

output: 520

10.Update the marks of an existing student.

dict1={

'Naina':480,

' ‘Parveen':520,

‘Parveen’:540

'Aaliyaa':380

}

print(dict1)

output: 540

11.Can I access a key using a value in a dictionary.

Yes, but Python dictionaries don't support direct reverse lookup (value → key).

12. Can I have duplicate values and keys in a dictionary? What happens if I wanted try to

duplicate key in a dictionary?

Yes, you **can** have **duplicate values** in a Python dictionary—there’s no restriction on

values. But for **keys**, **duplicates aren’t allowed**. If you try to define or assign the same

key twice, Python **overwrites** the previous entry with the new one.

13.Print all multiples of 17 using range. Numbers should start from -34 and end below 400.

print(list(range(-34,400,17)))

output: [-34, -17, 0, 17, 34, 51, 68, 85, 102, 119, 136, 153, 170, 187, 204, 221, 238, 255, 272, 289, 306, 323, 340, 357, 374, 391]