

ASSIGNMENT – 1

PYTHON FOR DATA SCIENCE

SRINIVAS THATTE
PES2UG22EC138

○ MCQS

1) What will be the output of the following code?

```
print(bool(0), bool(""), bool([]))
```

- a) True True True
- b) False False False
- c) True False True
- d) False True False

Answer : b) False False False

Explanation - 0, empty string "", and empty list [] are all **falsy** in Python.

2) Which of the following expressions returns True?

- a) 5 and 0
- b) 0 or []
- c) not ""
- d) bool(None)

Answer: c) not ""

Explanation: Empty string is falsy, so not "" → True.

○ DESCRIPTIVE QUESTION (7 MARKS)

1) **Given the tuple:**

```
data = (1, 2, 3, 4, 5)
```

Write Python code to demonstrate:

- I. Slicing to extract (2, 3, 4)
- II. Repetition to create (1, 2, 3, 4, 5, 1, 2, 3, 4, 5)
- III. Concatenation with (6, 7)
- IV. Membership check for 3
- V. Using built-in functions: len(), max(), and sum().

Answer :

```
data = (1, 2, 3, 4, 5)
```

I. Slicing

```
print(data[1:4])      # (2, 3, 4)
```

II. Repetition

```
print(data * 2)       # (1,2,3,4,5,1,2,3,4,5)
```

III. Concatenation

```
print(data + (6, 7))  # (1,2,3,4,5,6,7)
```

IV. Membership

```
print(3 in data)      # True
```

V. Functions

```
print(len(data))      # 5
```

```
print(max(data))      # 5
```

```
print(sum(data))      # 15
```

- 2) **Explain in detail the different basic data types in Python. Discuss each with examples, showing how to check the type of a variable. Also describe how type conversion (coercion) works with suitable examples.(5 marks)**

Answer : In Python, variables can store different **data types**, which define the kind of values they hold:

a) Integer (int) → Whole numbers

```
a = 10
```

```
print(type(a)) # <class 'int'>
```

b) Float (float) → Decimal numbers

```
b = 3.14
```

```
print(type(b)) # <class 'float'>
```

c) Boolean (bool) → Logical values (True, False)

```
flag = True
```

```
print(type(flag)) # <class 'bool'>
```

d) String (str) → Sequence of characters

```
name = "Python"
```

```
print(type(name)) # <class 'str'>
```

e) Complex (complex) → Numbers with real & imaginary parts

```
z = 2 + 3j
```

```
print(type(z)) # <class 'complex'>
```

Type Conversion (Coercion):

- 1) We can convert one type into another using built-in functions:

```
x = "100"
```

```
y = int(x) + 20 # converts string to int
```

```
print(y) # 120
```

