Q1: How will you demonstrate the following" Python is case sensitive programming language"? (2 marks)

Python treats identifiers with different letter casing as different names.

A simple way to demonstrate this is to use variables or functions that differ only by case.

Here’s an example:

```
name = "Alice"

Name = "Bob"


print(name)

print(Name)
```

Explanation:

- name and Name are treated as **two different variables**
- Output will be:
- Alice
- Bob

Another demonstration that shows an **error**:

```
number = 10

print(Number)   # Different case
```

This will raise:

```
NameError: name 'Number' is not defined
```

Conclusion:
name, Name, and NAME are not the same in Python — which proves Python is a case-sensitive programming language.

Q2: Provide 2 examples of implicit type casting? (2 marks)

Example 1: int → float during arithmetic

When an integer is used with a float, Python automatically converts the integer to a float.

```
a = 5     # int
b = 2.5   # float


c = a + b  # int automatically becomes float
print(c)
print(type(c))
```

Output

```
7.5
<class 'float'>
```

Example 2: int → complex during addition

When an integer is added to a complex number, Python converts the integer to complex automatically.

```
x = 3       # int
y = 4+2j    # complex
z = x + y
print(z)
print(type(z))
```

Output

```
(7+2j)
<class 'complex'>
```

Key point:

Implicit type casting happens automatically when Python promotes lower data types to higher compatible types (int → float → complex) to avoid data loss.

Q3: What is a complex data type in Python? (2 marks)

n Python, a **complex data type** represents **complex numbers**, which have **two parts**:

- a **real part**
- an **imaginary part** (with j or J as the imaginary unit)

Definition

A complex number in Python is written as:

a + bj

where:

- a → real part (int or float)
- b → imaginary part (int or float)
- j → represents $\sqrt{-1}$

Q4: What is the default parameter in user-defined function? (2 marks)

A default parameter (also called a default argument) in a user-defined Python function is a parameter that is given a value in the function definition. If the caller does not pass a value for that parameter, Python automatically uses the default value.

Default parameter = pre-assigned value used when no argument is supplied.

Example

```
def greet(name="Guest"): print("Hello,", name)

greet()        # uses default value "Guest"

greet("Alice")   # uses provided value
```

QA5: **Explain arithmetic operators which can be used with strings.**

In Python, only **two arithmetic operators** work with strings:

1) Addition operator + → String concatenation

It joins (concatenates) two strings to make one longer string.

```
s1 = "Hello"

s2 = "World"

result = s1 + " " + s2

print(result)
```

Output

```
Hello World
```

+ does **not** add values — it **joins strings together**.

---

2) Multiplication operator * → String repetition

It repeats a string **n** number of times (where n is an integer).

```
text = "Hi "

result = text * 3

print(result)
```

Output

```
Hi Hi Hi
```

Works only with **string × integer**, not string × string.

QA6: **Define significant features of NumPy library.**

Answer:

NumPy (Numerical Python) is a powerful Python library used for numerical and scientific computing. Its significant features include:

- Provides **N-dimensional array object (ndarray)**
- Supports **vectorized operations** (faster than Python lists)
- Offers **mathematical, logical, statistical, and linear algebra functions**
- Efficient **memory usage**
- Supports **broadcasting**
- Tools for **Fourier transform**
- Integration with **C/C++ and other languages**
- Foundation for **Pandas, SciPy, Scikit-learn, TensorFlow, etc.**

QA7: **Explain the difference between pivot table and cross table.?**

Pivot Table	**Cross Table (Crosstab)**
Summarizes data by applying aggregation (sum, mean, count, etc.)	Shows frequency or relationship between two categorical variables
Used for data analysis & summarization	Mainly used for contingency/frequency tables
Flexible – multiple aggregations possible	Usually counts occurrences
Example tool: Excel Pivot	Example tool: Pandas crosstab()

Example in Pandas:

- Pivot table → pivot_table()
- Crosstab → crosstab()

QA 8: Explain the random package of NumPy.

numpy.random is a sub-module used to generate random numbers.

Key uses:

- Generate random integers and floats
- Create random samples
- Generate random arrays
- Simulate probability distributions

Examples:

- rand() → random floats (0–1)
- randint() → random integers
- random_sample() → random samples
- choice() → random selection from array
- Supports distributions like:
  - normal
  - binomial
  - poisson
  - uniform

Used in:

- statistics
- simulations
- machine learning
- games
- data science

QA9: How to delete rows and columns from a DataFrame?

In Pandas, we delete rows or columns using drop().

Delete row

```
df = df.drop(2)        # delete row with index 2
```

Delete multiple rows

```
df = df.drop([1,3])
```

Delete column

```
df = df.drop('Age', axis=1)
```

Delete multiple columns

```
df = df.drop(['Age','Salary'], axis=1)
```

QA10: How to rank the rows of DataFrame?

We use **rank()** function in Pandas.

```
df['rank'] = df['Marks'].rank()
```

Types of ranking:

- average (default)
- min rank
- max rank
- dense rank

Example:

```
df['rank'] = df['Marks'].rank(method='dense', ascending=False)
```

This assigns rank based on values.