



```
5.  
x = 10  
def func():  
    x = 5  
    print(x)  
func()  
print(x)
```

### Section C: Short / Long Answer Questions

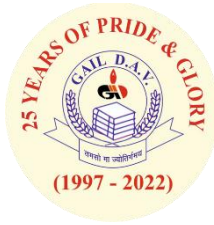
1. Explain the difference between **list and tuple** with examples.
2. What are **mutable and immutable data types**? Give examples.
3. Explain **types of errors in Python** (Syntax, Runtime, Logical) with examples.
4. What is **type conversion**? Explain implicit and explicit conversion.
5. Differentiate between **for loop and while loop** with examples.

### Section D: Programming Questions

1. Write a program to check whether a number is **prime or not**.
2. Write a program to **find factorial** of a number using a loop.
3. Write a program to **count vowels in a string**.
4. Write a program to create a list of numbers and print:
  - Maximum number
  - Minimum number
5. Write a program to **reverse a number** using a while loop.

### Section E: Case-Based / Application Questions

1. A student wants to store marks of 5 subjects and calculate the average.
  - Write a Python program using a list.
  - Display the average marks.
2. A shopkeeper wants to maintain a list of prices and find the total bill.
  - Write a Python program to calculate total and display highest price.



**Section A: Assertion – Reason**

Choose the correct option:

- A. Both Assertion and Reason are true, and Reason is the correct explanation
- B. Both Assertion and Reason are true, but Reason is not the correct explanation
- C. Assertion is true, Reason is false
- D. Assertion is false, Reason is true

1. **Assertion:** Lists are mutable in Python.

**Reason:** Elements of a list can be modified after creation.

2. **Assertion:** Tuples are faster than lists.

**Reason:** Tuples are immutable in nature.

3. **Assertion:** Dictionary keys must be unique.

**Reason:** Duplicate keys are not allowed in dictionaries.

4. **Assertion:** A tuple can contain mutable elements.

**Reason:** Lists can be stored inside tuples.

5. **Assertion:** Dictionary is an ordered collection.

**Reason:** Elements are stored in key-value pairs.

**Section B: Output-Based Questions**

1.

```
L = [10, 20, 30, 40]
```

```
L[2] = 100
```

```
print(L)
```

2.

```
t = (1, 2, 3)
```

```
print(t * 2)
```

3.

```
d = {"a": 1, "b": 2}
```

```
d["c"] = 3
```

```
print(len(d))
```

4.

```
L = [1, 2, 3]
```

```
print(L + [4, 5])
```

5.

```
d = {"x": 10, "y": 20}
```

```
print(d.get("z", 0))
```

### Section C: Short / Long Answer Questions

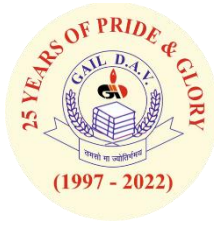
1. Differentiate between **List and Tuple** with at least 3 points.
2. Explain different **list methods** (append(), extend(), pop(), remove()) with examples.
3. What is a **dictionary**? Explain its characteristics.
4. Explain **tuple packing and unpacking** with example.
5. Write differences between **list and dictionary**.

### Section D: Programming Questions

1. Write a program to:
  - Create a list of numbers
  - Print sum and average
2. Write a program to **find the largest element in a tuple**.
3. Write a program to **count frequency of elements in a list**.
4. Write a program to create a **dictionary of student names and marks** and:
  - Display all students
  - Find highest marks
5. Write a program to **merge two dictionaries**.

### Section E: Case-Based Questions

1. A teacher wants to store student marks using a dictionary.
  - Input marks of 5 students
  - Display student with highest marks
2. A shop stores item prices in a list.
  - Add new items
  - Remove an item
  - Display final list



**GAIL DAV PUBLIC SCHOOL  
GAIL GAON, DIBIYAPUR, AURAIYA**

**Class XII**

**Subject: - Computer Science**

**Topic: Functions**

**Section A: Assertion – Reason**

Choose the correct option:

- A. Both Assertion and Reason are true, and Reason is the correct explanation
- B. Both Assertion and Reason are true, but Reason is not the correct explanation
- C. Assertion is true, Reason is false
- D. Assertion is false, Reason is true

1. **Assertion:** A function helps in code reusability.  
**Reason:** A function can be called multiple times in a program.
2. **Assertion:** A function without a return statement returns None.  
**Reason:** Every function must return a value in Python.
3. **Assertion:** Default arguments must be placed after non-default arguments.  
**Reason:** Python assigns values to parameters from left to right.
4. **Assertion:** Lambda functions can have multiple expressions.  
**Reason:** Lambda functions are anonymous functions.
5. **Assertion:** Local variables cannot be accessed outside the function.  
**Reason:** Their scope is limited to the function block.

**Section B: Output-Based Questions**

1.  
def func(x, y=5):  
 return x + y  
print(func(3))

2.  
def display():  
 print("Hello")  
x = display()  
print(x)

3.  
def f(a, b):  
 print(a, b)  
f(b=2, a=5)

4.  
x = 10  
def test():  
 x = 5  
 print(x)  
test()  
print(x)

5.  
square = lambda x: x\*x  
print(square(4))

### Section C: Short / Long Answer Questions

1. Define a **function**. Explain its advantages.
2. Differentiate between **formal parameters and actual parameters**.
3. Explain the types of arguments in Python:
  - Default arguments
  - Keyword arguments
  - Positional arguments
4. What is the difference between **local and global variables**?
5. Explain **lambda functions** with examples.

### Section D: Programming Questions

1. Write a function to **find the sum of two numbers** and return the result.
2. Write a function to check whether a number is **even or odd**.
3. Write a function to **calculate factorial** using recursion.
4. Write a function that accepts a string and returns the **number of vowels**.
5. Write a function using default argument to calculate **simple interest**.

### Section E: Case-Based Questions

1. A teacher wants to calculate grades of students using functions.
  - Write a function that takes marks as input and returns grade:
    - $\geq 90 \rightarrow A$
    - $\geq 75 \rightarrow B$
    - $\geq 50 \rightarrow C$
    - $< 50 \rightarrow \text{Fail}$
2. A shop uses a billing system:
  - Write a function to calculate total cost of items.
  - Add GST (default 18%) using default argument.