Python Programming

Unit-wise Previous Year Question Bank

Subject: Python Programming

Codes: 1323203 / DI02032011 (Diploma Engineering – Sem 2)

Papers covered: Summer 2023 · Winter 2023 · Summer 2024 · Winter 2024 · Summer 2025 · Winter 2025

Questions grouped unit-wise with exam name, question number & marks

---

Notation Guide

Paper codes	:	Su'23 = Summer 2023 (1323203) · W'23 = Winter 2023 (1323203) · Su'24 = Summer 2
		W'24 = Winter 2024 (1323203) · Su'25 = Summer 2025 (DI02032011) · W'25 = Winter
Exam format	:	Q.X(a) [3 M]; Q.X(b) [4 M]; Q.X(c) [7 M]; Q.X(c) [OR] [7 M]
Green	:	Appeared as main / compulsory question
Purple	:	Appeared as OR (alternative) question

Question-bank Statistics

Unit 1 (Problem Solving using Flowchart and Algorithm)	:	14 questions
Unit 2 (Introduction to Python)	:	21 questions
Unit 3 (Flow of Control)	:	31 questions
Unit 4 (Functions)	:	31 questions
Unit 5 (Dictionary, List, Set, String and Tuple)	:	39 questions
Total	:	**136 unique questions** across 6 papers

Contents

Unit 1 | Problem Solving using Flowchart and Algorithm

1.1 Flowchart & Algorithm — 3-mark questions

1. **Define flowchart and list any four symbols used in flowchart with their purpose** [3 M]
   - *Appeared in:* W'24 – Q.1(a); Su'25 – Q.1(a)
2. **Define algorithm. What are the advantages of Algorithm?** [3 M]
   - *Appeared in:* Su'23 – Q.1(a)
3. **List the importance of flowchart and algorithm** [3 M]
   - *Appeared in:* Su'24 – Q.1(a)
4. **Define Algorithm. Write algorithm to find area of a circle** [3 M]
   - *Appeared in:* W'25 – Q.1(a)
5. **Write the difference between flowchart and algorithm** [3 M]
   - *Appeared in:* Su'23 – Q.2(a) [OR]

1.2 Flowchart & Algorithm — 4-mark questions

6. **State rules for problem solving using flowchart. Design flowchart to find simple interest** [4 M]
   - *Appeared in:* Su'23 – Q.1(b); W'25 – Q.1(b)
7. **Define Algorithm. Design algorithm for finding maximum of three numbers** [4 M]
   - *Appeared in:* W'23 – Q.1(b)
8. **Draw flowchart to find whether the entered number is even or odd** [4 M]
   - *Appeared in:* Su'24 – Q.1(b)
9. **Prepare algorithms for the following: (1) To find maximum of two given numbers (2) To find sum of two given numbers** [4 M]
   - *Appeared in:* Su'25 – Q.1(b)

1.3 Flowchart & Algorithm — 7-mark questions

10. **Design flowchart to calculate sum of first twenty even natural numbers** [7 M]
    - *Appeared in:* W'24 – Q.1(c)
11. **Create algorithm to print odd numbers between 1 to 20** [7 M]
    - *Appeared in:* W'24 – Q.1(c) [OR]

1.4 Pseudocode — 3-mark questions

12. **Write pseudocode to check if the entered number is positive or negative** [3 M]

- *Appeared in:* W'23 – Q.1(a)

13. **Define pseudocode. Write pseudocode to find the smallest of two numbers** [3 M]
    - *Appeared in:* Su'23 – Q.2(a)

1.5 Pseudocode — 4-mark questions

14. **Define pseudocode. Write pseudocode to find the largest of three numbers x, y, z** [4 M]
    - *Appeared in:* Su'24 – Q.2(b)

Unit 2 | Introduction to Python

2.1 Python Features & Applications — 3-mark questions

1. **List the features of Python** [3 M]
    - *Appeared in:* Su'25 – Q.2(a); W'25 – Q.2(a)

2. **List the applications of Python** [3 M]
    - *Appeared in:* Su'25 – Q.2(a) [OR]; W'25 – Q.2(a) [OR]

2.2 Data Types, Variables & Operators — 3-mark questions

3. **Describe data types in Python with examples** [3 M]
    - *Appeared in:* W'23 – Q.2(a)

4. **List out rules for defining variables in Python** [3 M]
    - *Appeared in:* W'23 – Q.2(a) [OR]

5. **Discuss membership operator of Python with example** [3 M]
    - *Appeared in:* W'24 – Q.2(a)

6. **Write a short note on assignment operator in Python** [3 M]
    - *Appeared in:* W'24 – Q.2(a) [OR]

2.3 Data Types & Operators — 4-mark questions

7. **List various data types in Python and explain any three with example** [4 M]
    - *Appeared in:* W'24 – Q.1(b)

8. **Develop Python code to read three numbers and find their average** [4 M]
    - *Appeared in:* Su'23 – Q.2(b)

9. **Determine the output of the following Python code snippet involving arithmetic and comparison operators (e.g. `x*y`, `x**y`, `x//y`, `x%y`, `x>y`)** [4 M]
    - *Appeared in:* Su'23 – Q.2(b) [OR]; Su'24 – Q.2(b) [OR]

2.4 Programs using Operators & Data Types — 7-mark questions

10. **List assignment operators in Python and build Python code to demonstrate any three** [7 M]
    - *Appeared in:* Su'23 – Q.1(c)

11. **List data types in Python and develop a program to identify any three** [7 M]
    - *Appeared in:* Su'23 – Q.1(c) [OR]

12. **Develop Python code to convert temperature from Celsius to Fahrenheit** [7 M]
    - *Appeared in:* W'23 – Q.1(c)

13. **List all comparison operators and explain each with Python code example** [7 M]
    - *Appeared in:* W'23 – Q.1(c) [OR]

14. **List all logical operators and explain each with Python code example** [7 M]
    - *Appeared in:* Su'24 – Q.1(c)

15. **Develop a program to calculate simple interest and compound interest** [7 M]
    - *Appeared in:* Su'24 – Q.1(c) [OR]

16. **Create a program to calculate total and average marks of a student based on four subject marks** [7 M]
    - *Appeared in:* W'24 – Q.2(c)

17. **Develop code to find square and cube of a given number input by user** [7 M]
    - *Appeared in:* W'24 – Q.2(c) [OR]

18. **List different data types in Python and explain all data types with examples** [7 M]
    - *Appeared in:* Su'25 – Q.2(c)

19. **List different types of operators in Python and explain any two types with example** [7 M]
    - *Appeared in:* Su'25 – Q.2(c) [OR]

20. **List different data types in Python. Explain any two data types with example** [7 M]
    - *Appeared in:* W'25 – Q.1(c)

21. **Explain Arithmetic operators and Relational operators with Python code example** [7 M]
    - *Appeared in:* W'25 – Q.1(c) [OR]

Unit 3 | Flow of Control

3.1 Jump Statements (break & continue) — 3-mark questions

1. **Describe the use of `break` statement with necessary examples** [3 M]
   - *Appeared in:* Su'23 – Q.3(a)

2. **Describe the use of `continue` statement with necessary examples** [3 M]
   - *Appeared in:* Su'23 – Q.3(a) [OR]; Su'24 – Q.2(a) [OR]

3.2 Selection Statements — 3-mark questions

3. **Develop Python code to identify if the scanned number is even or odd** [3 M]
   - *Appeared in:* Su'23 – Q.4(a)

4. **Create Python code to check if a number is positive or negative** [3 M]
   - *Appeared in:* Su'23 – Q.4(a) [OR]

5. **Explain if-elif-else statement with flowchart and suitable example** [3 M]
   - *Appeared in:* W'24 – Q.3(a)

3.3 Loop Statements — 3-mark questions

6. **Develop Python program to print numbers 1 to 10 using loops** [3 M]
   - *Appeared in:* W'23 – Q.3(a)

7. **Develop Python code to find odd and even numbers from 1 to N using loops** [3 M]
   - *Appeared in:* W'23 – Q.3(a) [OR]

8. **Write Python program to print odd numbers between 1 to 20 using loops** [3 M]
   - *Appeared in:* Su'24 – Q.3(a)

9. **Write Python program to find sum of numbers from 1 to 100** [3 M]
   - *Appeared in:* Su'24 – Q.3(a) [OR]

10. **Explain nested loop with a suitable example** [3 M]
    - *Appeared in:* W'24 – Q.3(a) [OR]

3.4 Selection Statements — 4-mark questions

11. **Explain if...else statement with suitable example** [4 M]
    - *Appeared in:* Su'23 – Q.3(b); Su'25 – Q.2(b); W'25 – Q.2(b)

12. **Explain Nested if statement in Python with python code example** [4 M]
    - *Appeared in:* W'23 – Q.2(b); Su'24 – Q.3(b); W'25 – Q.2(b) [OR]

3.5 Loop Statements — 4-mark questions

13. **Explain `for` loop statement with example** [4 M]
    - *Appeared in:* Su'23 – Q.3(b) [OR]; W'23 – Q.2(b) [OR]; W'24 – Q.2(b) [OR]; Su'25 – Q.3(b)

14. **Explain the need for `continue` and `break` statements in Python** [4 M]
    - *Appeared in:* W'24 – Q.2(b)

15. **Explain `while` loop with example** [4 M]
    - *Appeared in:* Su'25 – Q.2(b) [OR]

16. **Explain `break` statement with example** [4 M]
    - *Appeared in:* Su'25 – Q.3(b) [OR]

17. **Give differences between `for` loop and `while` loop. Develop a Python program to print numbers 1 to 25 using `for` loop** [4 M]
    - *Appeared in:* W'25 – Q.3(b)

18. **Develop Python program to find if the entered number is prime or not** [4 M]
    - *Appeared in:* W'25 – Q.3(b) [OR]

3.6 Pattern Programs — 4-mark questions

19. **Develop Python program to print the following star pattern:** [4 M]

```
*
* *
* * *
* * * *
* * * * *
```

    - *Appeared in:* W'23 – Q.3(b)

20. **Write Python program to print the following number triangle pattern:** [4 M]

```
1   2 3   4 5 6   7 8 9 10
```

    - *Appeared in:* Su'24 – Q.3(b) [OR]

3.7 Selection, Loop & Pattern Programs — 7-mark questions

21. **Write Python code to show whether the entered number is prime or not** [7 M]
    - *Appeared in:* Su'23 – Q.2(c)

22. **Write Python code to display the following patterns: (A) number triangle (B) inverted star triangle** [7 M]
    - *Appeared in:* Su'23 – Q.2(c) [OR]

23. **Explain different types of selection/decision making flow-of-control structures with example** [7 M]
    - *Appeared in:* W'23 – Q.2(c)

24. Describe `break` and `continue` statements in Python in brief [7 M]
    - *Appeared in:* W'23 – Q.2(c) [OR]
25. Explain `while` loop with syntax, flowchart and Python code example [7 M]
    - *Appeared in:* Su'24 – Q.2(c)
26. Explain if-elif-else ladder with syntax, flowchart and Python code [7 M]
    - *Appeared in:* Su'24 – Q.2(c) [OR]
27. Create program to display the following patterns: (A) `&` character triangle (B) number triangle (1 / 1 2 / 1 2 3 / 1 2 3 4 / 1 2 3 4 5) [7 M]
    - *Appeared in:* Su'25 – Q.1(c)
28. Develop Python programs: (1) To check if entered number is odd or even (2) To find average of numbers from 1 to n, where n is taken from user [7 M]
    - *Appeared in:* Su'25 – Q.1(c) [OR]
29. Create program to display the following patterns: (A) `@` inverted character triangle (B) squared number series (1 / 1 4 / 1 4 9 / 1 4 9 16) [7 M]
    - *Appeared in:* W'25 – Q.2(c)
30. Explain `for` loop in Python with its syntax, flowchart and Python code example [7 M]
    - *Appeared in:* W'25 – Q.2(c) [OR]
31. Create program to display the following pattern: A / AB / ABC / ABCD / ABCDE [7 M]
    - *Appeared in:* W'24 – Q.3(c) [OR]

Unit 4 | Functions

4.1 Functions & Scope of Variables — 3-mark questions

1. Explain local variable with example [3 M]
    - *Appeared in:* Su'25 – Q.3(a)
2. Explain global variable with example [3 M]
    - *Appeared in:* Su'25 – Q.3(a) [OR]
3. List Python standard library mathematical functions [3 M]
    - *Appeared in:* W'23 – Q.4(a); Su'24 – Q.4(a) [OR]
4. Describe Python math module with proper Python code example [3 M]
    - *Appeared in:* Su'24 – Q.4(a)
5. Describe built-in functions `max()`, `input()`, `pow()` with example [3 M]

- *Appeared in:* W'24 – Q.4(a)

6. **Explain random module with various functions** [3 M]
   - *Appeared in:* W'24 – Q.4(a) [OR]
7. **Explain built-in functions in Python** [3 M]
   - *Appeared in:* W'23 – Q.4(a) [OR]; Su'24 – Q.4(b) [OR]

4.2 User-defined Functions — 4-mark questions

8. **Define function. Explain user-defined function with suitable example** [4 M]
   - *Appeared in:* Su'23 – Q.4(b)
9. **Explain how to define and call a user-defined function with suitable example** [4 M]
   - *Appeared in:* W'24 – Q.3(b)
10. **Explain return statement in function handling** [4 M]
    - *Appeared in:* W'24 – Q.3(b) [OR]
11. **Develop Python program using user-defined function to find factorial of the number taken from user** [4 M]
    - *Appeared in:* Su'25 – Q.4(b); W'25 – Q.4(b) [OR]
12. **Develop Python program using user-defined function to check if the number taken from user is prime** [4 M]
    - *Appeared in:* Su'25 – Q.4(b) [OR]
13. **Create a user-defined function to print the Fibonacci series of 0 to N numbers (N as argument)** [4 M]
    - *Appeared in:* W'25 – Q.4(b)

4.3 Scope of Variables — 4-mark questions

14. **Explain local and global variables with suitable examples** [4 M]
    - *Appeared in:* Su'23 – Q.4(b) [OR]
15. **Write Python program that explains the scope of a variable** [4 M]
    - *Appeared in:* Su'24 – Q.4(b)

4.4 Modules & Built-in Functions — 4-mark questions

16. **Explain math module and random module with example** [4 M]
    - *Appeared in:* Su'23 – Q.5(b)
17. **Explain usage of built-in functions: `print()`, `min()`, `sum()`, `input()`** [4 M]
    - *Appeared in:* Su'23 – Q.5(b) [OR]
18. **Explain Module in Python with python code example** [4 M]
    - *Appeared in:* W'23 – Q.4(b)
19. **Describe Python math module with python code example** [4 M]

- *Appeared in:* W'23 – Q.4(b) [OR]

20. Explain Random module in Python [4 M]

- *Appeared in:* W'25 – Q.5(b)

21. Explain statistics module in Python [4 M]

- *Appeared in:* W'25 – Q.5(b) [OR]

4.5 User-defined Functions & Programs — 7-mark questions

22. Create user-defined function to print Fibonacci series 0 to N [7 M]

- *Appeared in:* Su'23 – Q.3(c); W'24 – Q.4(c) [OR]

23. Write Python code to determine if a number is Armstrong or palindrome using user-defined function [7 M]

- *Appeared in:* Su'23 – Q.3(c) [OR]

24. Create user-defined function to find factorial of a given number [7 M]

- *Appeared in:* W'23 – Q.3(c); W'24 – Q.3(c)

25. Explain local and global variables with examples; explain the concept of scope of variable [7 M]

- *Appeared in:* W'23 – Q.3(c) [OR]; W'23 – Q.4(c) [OR]; W'25 – Q.3(c)

26. Write Python program to determine if a number is Armstrong or palindrome using user-defined function (number passed as argument) [7 M]

- *Appeared in:* W'23 – Q.4(c); Su'24 – Q.3(c)

27. Write program using a function that reverses the entered value [7 M]

- *Appeared in:* Su'24 – Q.3(c) [OR]

28. Create user-defined function which prints cube of all odd numbers between 1 to 7 [7 M]

- *Appeared in:* W'24 – Q.4(c)

29. List functions of math module in Python and develop a Python program demonstrating math module functions [7 M]

- *Appeared in:* Su'25 – Q.3(c)

30. List functions of statistics module in Python and develop a Python program demonstrating statistics module functions [7 M]

- *Appeared in:* Su'25 – Q.3(c) [OR]

31. List Python standard library mathematical functions and explain any three in brief [7 M]

- *Appeared in:* W'25 – Q.3(c) [OR]

Unit 5 | Dictionary, List, Set, String and Tuple

5.1 List — 3 & 4-mark questions

1. **Write Python code to swap two elements in a list** [3 M]
   - *Appeared in:* Su'23 – Q.5(a); W'23 – Q.5(a); Su'24 – Q.5(a); Su'25 – Q.5(a) [OR]
2. **Write Python code to find sum of all elements in a list** [3 M]
   - *Appeared in:* Su'23 – Q.5(a) [OR]; W'23 – Q.5(a) [OR]; Su'24 – Q.5(a) [OR]
3. **Define list and explain how it is created in Python** [3 M]
   - *Appeared in:* W'24 – Q.5(a) [OR]
4. **Find the smallest number in the given list: `List1 = [21, 6, 3, 18, 12, 15]`** [3 M]
   - *Appeared in:* Su'25 – Q.5(a)
5. **List any six built-in functions of List** [3 M]
   - *Appeared in:* W'25 – Q.4(a)
6. **Develop Python program to find multiplication of all elements in the list** [3 M]
   - *Appeared in:* W'25 – Q.5(a)
7. **Develop code to create a nested list and display its elements** [4 M]
   - *Appeared in:* W'23 – Q.3(b) [OR]
8. **Explain nested list by giving example** [4 M]
   - *Appeared in:* W'23 – Q.5(b)
9. **Explain indexing and slicing operations in Python list** [4 M]
   - *Appeared in:* W'23 – Q.5(b) [OR]
10. **Discuss list functions: `len()`, `sum()`, `sort()`, `index()`** [4 M]
    - *Appeared in:* W'24 – Q.4(b) [OR]

5.2 String — 3 & 4-mark questions

11. **List any six built-in functions of string** [3 M]
    - *Appeared in:* W'25 – Q.3(a)
12. **Explain any one String Operation with example** [3 M]
    - *Appeared in:* W'25 – Q.3(a) [OR]
13. **Explain string methods: `count()`, `upper()`, `replace()` with examples** [3 M]
    - *Appeared in:* W'24 – Q.5(a)
14. **Explain indexing in string with example** [3 M]
    - *Appeared in:* Su'25 – Q.4(a) [OR]
15. **Develop Python program to count and display number of vowels in a string** [3 M]
    - *Appeared in:* W'25 – Q.4(a) [OR]
16. **Explain slicing of string with suitable example** [4 M]

- *Appeared in:* W'24 – Q.4(b)

17. Write Python program to check if a substring is present in a given string [4 M]
- *Appeared in:* Su'24 – Q.5(b)

18. Develop Python program to demonstrate any 4 built-in functions for string [4 M]
- *Appeared in:* Su'25 – Q.5(b)

5.3 Tuple — 3 & 4-mark questions

19. Explain concept of nested tuple with example [3 M]
- *Appeared in:* Su'25 – Q.4(a)

20. Explain tuple operations with example [4 M]
- *Appeared in:* W'24 – Q.5(b)

21. Develop Python program to demonstrate any 4 built-in functions for tuple [4 M]
- *Appeared in:* Su'25 – Q.5(b) [OR]

5.4 Set & Dictionary — 4-mark questions

22. Write program to demonstrate set functions and operations [4 M]
- *Appeared in:* Su'24 – Q.5(b) [OR]

23. Explain dictionary built-in functions and methods [4 M]
- *Appeared in:* W'24 – Q.5(b) [OR]

24. Develop a Python program to create a Dictionary with the roll number, name and marks of n students; display the names of students who have scored marks above 70 [4 M]
- *Appeared in:* W'25 – Q.5(a) [OR]

5.5 List — 7-mark questions

25. List various list operations and explain any three with example [7 M]
- *Appeared in:* Su'23 – Q.4(c) [OR]

26. Explain List Methods and its built-in Functions [7 M]
- *Appeared in:* Su'24 – Q.4(c)

27. Explain List operations with examples [7 M]
- *Appeared in:* Su'25 – Q.4(c)

28. Develop Python code to create a list of prime numbers and a list of non-prime numbers in the range 1 to 50 [7 M]
- *Appeared in:* W'24 – Q.5(c) [OR]

29. Explain Indexing and Slicing in List with example [7 M]
- *Appeared in:* W'25 – Q.5(c) [OR]

5.6 String — 7-mark questions

30. **List various string operations and explain any three with example** [7 M]
    - *Appeared in:* Su'23 – Q.4(c)

31. **Explain string operations with examples** [7 M]
    - *Appeared in:* W'23 – Q.5(c)

32. **Write Python program to count and display vowels, consonants, uppercase letters and lowercase letters in a string** [7 M]
    - *Appeared in:* Su'24 – Q.4(c) [OR]

5.7 Tuple — 7-mark questions

33. **Write Python code to demonstrate tuple functions and operations** [7 M]
    - *Appeared in:* Su'23 – Q.5(c)

34. **Explain tuple in brief with necessary example** [7 M]
    - *Appeared in:* W'23 – Q.5(c) [OR]

35. **Explain tuple Operations, Functions and Methods** [7 M]
    - *Appeared in:* Su'24 – Q.5(c)

36. **Explain tuple operations with examples** [7 M]
    - *Appeared in:* Su'25 – Q.4(c) [OR]; W'25 – Q.4(c) [OR]

5.8 Set — 7-mark questions

37. **List set functions and operations; develop a Python program to demonstrate set functions and operations** [7 M]
    - *Appeared in:* Su'23 – Q.5(c) [OR]; Su'25 – Q.5(c); W'25 – Q.4(c)

38. **Develop code for two sets and perform union, intersection, difference, and symmetric difference operations** [7 M]
    - *Appeared in:* W'24 – Q.5(c)

5.9 Dictionary — 7-mark questions

39. **Write program to demonstrate dictionary functions and operations** [7 M]
    - *Appeared in:* Su'24 – Q.5(c) [OR]; Su'25 – Q.5(c) [OR]

40. **Explain Dictionary Built-in Functions with examples** [7 M]
    - *Appeared in:* W'25 – Q.5(c)