

# Python Programming Laboratory (MR25)

## Viva Questions

### Task 1 (Type Casting and Operators)

1. Define **variables** in Python. What are the **rules for naming variables**? Give examples.
2. Discuss different **data types** in Python with examples.
3. Explain **type casting** in Python with suitable code examples.
4. Discuss **relational operators** in Python with suitable programs.
5. With examples, illustrate the use of **assignment** and **augmented assignment operators**.
6. Give examples of `print()` function using “**str.format()**”, and “**f-strings**”.

### Task 2 (Control Structures)

1. Write a Python program using “**if statement**” to check whether a number is positive.
2. Which statement executes if **condition** is false?
3. Why is **elif** used?
4. Example of short-hand “**if else**” statement.
5. How do you write **multi-line comments**?
6. What is used to define **code blocks** in Python?

### Task 3 (Loops)

1. What happens if condition in “**while**” loop never becomes false?
2. Write a program to print all **even** and **odd** numbers between 1 and 20 using loops.
3. What is the purpose of **else clause** for a loop? Explain how else works with **for** and **while loops**, with example.
4. Which **keyword** is used to **exit a loop**?
5. Which **keyword skips** current iteration?
6. What does `pass` do?

## Task 4 (Sequence Types)

1. What is **string indexing**?
2. Write the output of "Hello"[1].
3. Which method is used to **remove whitespace** from both ends of a string?
4. Which keyword is used in **list comprehension**?
5. Differentiate between **append()** and **extend()** methods in lists.
6. Why are **tuples** faster than **lists** in Python?

## Task 5 (Mapping Type and Sets)

1. Write a **dictionary comprehension** to create a dictionary of numbers 1 to 5 with their cubes.
2. How is a **dictionary** different from a **list** in Python?
3. Name one method used with **dictionaries** to get all keys.
4. Write the difference between **remove()** and **discard()** in sets.
5. Write a program to remove duplicate elements from a **list** using a **set**.
6. Write any four methods of **sets**.

## Task 6 (Functions)

1. Which **type of argument** is passed in the same order as defined?
2. What are **variable-length arguments**?
3. Which type of argument provides a **predefined value** if no value is passed?
4. What happens if you try to access a **local variable** outside the function?
5. What is the difference between **local** and **global** variables?
6. If a **local and global variable** have the same name, which one is preferred **inside the function**?

## **Task 7 (Lambda and Recursive Functions)**

1. Give one difference between **def** function and a **lambda** function.
2. How many arguments does **map()** take?
3. Write a **filter()** to extract even numbers from [1,2,3,4].
4. What does **reduce()** do?
5. What is the essential part of a **recursive function**?
6. Write a **recursive function** to calculate the sum of first n natural numbers.

## **Task 8 (Modules and Packages)**

1. Differentiate between **import math** and **from math import sqrt**.
2. Write the steps to create and use a **user-defined module**.
3. Write two uses of the **random module**.
4. What is the advantage of using **modules** in Python?
5. How is a **package** different from a **module**?
6. Why are **packages** important in Python?

## **Task 9 (File Operations and Regular Expressions)**

1. How do **json.dump()** and **json.dumps()** differ?
2. List any five file methods.
3. What is the purpose of **seek()** method? Give an example.
4. Write a Python statement to **open a file data.csv in append mode**.
5. Differentiate between **re.search()** and **re.match()**.
6. Write a **regex pattern** to validate an Indian mobile number starting with 7, 8, or 9 and having 10 digits.

## Task 10 (Exceptions)

1. Differentiate between **syntax error** and **runtime error**.
2. Give an example of **raising an exception** manually.
3. What happens if **except** clause is written without any Exception type?
4. What is the purpose of the **else block** in exception handling?
5. Name two **built-in exceptions** in Python.
6. What is the use of the **finally block**?

## Task 11 (Object Oriented Programming)

1. Differentiate between **class attribute** and **instance attribute**.
2. How is **self** used inside a class? Give an example.
3. What is the **default return type** of a **method** if return is not used?
4. Explain the **role of methods in classes** with an example.
5. What is the purpose of the **\_\_init\_\_()** method in a class?
6. What is the advantage of **inheritance**?
7. What **type of inheritance** allows a class to inherit from **more than one parent**?
8. Which symbols are used for **private** and **protected attributes** in Python?
9. Difference between **encapsulation** and **polymorphism**.
10. Differentiate between **method overloading** and **method overriding**.

## Task 12 (GUI programming)

1. Which **Python library** is used to create the GUI?
2. What is a **widget**?
3. List any **5 building blocks** of GUI.
4. What is the purpose of **root = tk.Tk()**?
5. What is **root.mainloop()**?