Assignment 1

1 Mark

1. Who developed Python and where?

A. Dennis Ritchie at Bell Labs
B. Guido van Rossum at the National Research Institute for Mathematics and Computer Science
C. James Gosling at Sun Microsystems
D. Tim Peters at MIT

Answer:
B. Guido van Rossum at the National Research Institute for Mathematics and Computer Science

2. What does **late binding** in Python mean?

A. Variables are bound at compile time

B. Variables are declared before use

C. Python cannot support runtime changes

D. Functions are looked up by name during runtime

Answer:

D. Functions are looked up by name during runtime

5-Marks

1. **Describe the main features of Python as a programming language.**

   Answer:

   Python is a **multi-paradigm language** supporting functional, object-oriented, and structural programming. It features **dynamic typing** with **runtime type checking**, **late binding** for resolving functions during execution, and **automatic memory management** through **reference counting**. Being **cross-platform**, it runs on Windows, macOS, and Linux, and offers **extensive libraries** for **data analysis, AI, ML**, and more.

2. **List and explain Python's sequence data types with examples.**

   Answer :

   Python provides three main sequence data types: strings, lists, and tuples. Strings are immutable sequences of characters, meaning their content cannot be changed after creation. For example, s = "Python" allows access to characters using indexing like s[0] which returns "P". Lists are mutable and ordered collections that can store heterogeneous data, and their elements can be modified, added, or removed. For instance, lst = [1, 2, 3] and lst.append(4) updates it to [1, 2, 3, 4]. Tuples, on the other hand, are immutable ordered collections used when data should remain constant, such as tup = (1, 2, 3), where attempting to modify an element raises an error. These sequence types are widely used for storing and manipulating ordered data in Python.

7-Marks

1. **Explain indexing and slicing operations on Python sequences with examples.**

   Answer:

   Indexing and slicing are powerful techniques used to access and manipulate elements within sequence data types like strings, lists, and tuples. Indexing allows accessing individual elements using their position, where positive indexes start from 0 (beginning) and negative indexes start from -1 (end).

   For example:

```
s = "Python"
print(s[0])   # Output: P
print(s[-1])  # Output: n
```

   Slicing, on the other hand, is used to extract a subsequence by specifying a start, end, and an optional step in the format [start:end:step]. The start index is inclusive, while the end index is exclusive.

   For example:

```
s = "PythonProgramming"
print(s[0:6])    # Output: Python
print(s[6:])     # Output: Programming
print(s[::-1])   # Output: gnimargorPnohtyP
```

   Thus, indexing retrieves single elements, whereas slicing extracts multiple elements or subsequence's efficiently. These operations make Python sequences flexible and easy to manipulate.

2. **Write a Python program to find the second largest element in a list and explain the logic.**
   Answer:

```
numbers = [10, 25, 7, 36, 18, 36]

# Convert list to a set to remove duplicates, then sort it
unique_numbers = sorted(set(numbers))

# Get the second largest element
second_largest = unique_numbers[-2]

print("Second Largest Element:", second_largest)
```

   Explanation:

1. **Remove duplicates → set(numbers) ensures only unique elements remain.**
2. **Sort the list → sorted() arranges numbers in ascending order.**
3. **Access the second largest → Using negative indexing [-2] retrieves the second last element.**
4. **Efficient approach → Avoids manual comparisons and works for any list size.**