Multiple Choice Questions (1 Mark Each)

1. Which of the following is a mutable data type in Python?

a) String

b) Tuple

c) Integer

d) List

Answer: d) List. Lists are mutable, meaning their elements can be changed after they are created. Strings, tuples, and integers are immutable.

2. What is the output of the following Python code snippet?

Python

```
x = 5
y = 2
print(x // y)
```

a) 2.5

b) 2

c) 1

d) Error

Answer: b) 2. The // operator performs floor division, which divides and rounds down to the nearest whole number.

Descriptive Questions (5 or 7 Marks Each)

1. Explain the key differences between a List and a Tuple in Python. Provide a code example for each to illustrate a situation where one would be preferred over the other. (5 Marks)

Answer:

The primary difference between a List and a Tuple in Python lies in their mutability.

Mutability:

Lists are mutable. This means you can change their content after they are created—you can add, remove, or modify elements. They are defined using square brackets [].

Tuples are immutable. Once a tuple is created, you cannot change its content. Elements cannot be added, removed, or modified. They are defined using parentheses ().

Performance:

Tuples are generally more memory-efficient and faster to process than lists because they are immutable. The fixed nature of tuples allows Python to perform internal optimizations.

Usage:

Use a List when you have a collection of items that might need to change during the program's execution. For example, a list of students registered for a course, where students might be added or removed.

Use a Tuple when you have a collection of items that should not change. This ensures data integrity. For example, storing the coordinates (x, y) of a point, or the RGB values of a color.

Code Examples:

List Example (Preferred for data that changes):

Python

```
# A list of tasks for a to-do list
tasks = ['Buy groceries', 'Pay bills', 'Call the bank']
print("Initial tasks:", tasks)
```

Modifying the list

tasks.append('Schedule appointment') # Adding an item

tasks.remove('Pay bills') # Removing an item

tasks[0] = 'Buy fresh vegetables' # Changing an item

print("Updated tasks:", tasks)

Tuple Example (Preferred for constant data):

Python

A tuple to store fixed configuration settings like server IP and port

server_config = ('192.168.1.1', 8080)

print(f"Connecting to server at IP: {server_config[0]} on Port: {server_config[1]}")

Attempting to change the tuple will result in an error, ensuring data integrity

server_config[0] = '127.0.0.1' # This line would raise a TypeError

2. Describe the concepts of variables and different built-in data types in Python (Integers, Floats, Strings, Booleans). Provide examples for each. (7 Marks)

Answer:

In Python, a variable is a symbolic name that acts as a reference or a pointer to an object stored in memory. Once an object is assigned to a variable, you can refer to the object by that name. Python is dynamically typed, meaning you don't need to declare the type of a variable before assigning a value to it.

Python has several built-in data types that are used to define the operations possible on them and the storage method for each of them. The fundamental types are:

1. Integers (int):

Description: Integers are whole numbers, both positive and negative, without any decimal point. They can be of any length, limited only by the memory available.

Example:

Python

```
student_count = 30
temperature = -5
print(f"Number of students: {student_count}, Type: {type(student_count)}")
```

2. Floating-Point Numbers (float):

Description: Floats represent real numbers and are written with a decimal point dividing the integer and fractional parts. They can also represent scientific notation.

Example:

Python

```
pi_value = 3.14159
account_balance = 150.75
print(f"Value of Pi: {pi_value}, Type: {type(pi_value)}")
```

3. Strings (str):

Description: A string is a sequence of characters enclosed within single quotes ('...'), double quotes ("..."), or triple quotes ('''...''' or """..."""). Strings are immutable, meaning the characters of a string cannot be changed after it is created.

Example:

Python

```
course_name = "Python for Data Science"
greeting = 'Hello, World!'
print(f"Course: {course_name}, Type: {type(course_name)}")
```

4. Booleans (bool):

Description: The Boolean type has two possible values: True or False. It is used to represent truth values. Boolean values are often the result of comparison operations.

Example:

Python

```
is_enrolled = True
is_complete = False
print(f"Is student enrolled? {is_enrolled}, Type: {type(is_enrolled)}")
print(f"Is 5 > 3? {5 > 3}") # This expression evaluates to True
```