ASSIGNMENT

Q1. How do you comment code in Python? What are the different types of comments?

ANS: Single-line Comments: These start with # and continue until the end of the line. For example: x = 5

Multi-line Strings as Comments: While not traditional comments, multi-line strings (triple-quoted strings) are often used as a workaround for multi-line comments. These strings are not assigned to any variable and hence are effectively treated as comments. For example:

"""This is a multi-line comment.

It spans multiple lines without the need for a # on each line.

"""

Docstring Comments: These are special comments used to document functions, classes, methods, or modules. They are enclosed in triple quotes (""" """) and can span multiple lines. Unlike regular comments, docstrings are stored as an attribute of the object they document and can be accessed programmatically. For example:

```
def my_function():
    """
    This function does something.
    """
    Pass
```

Q2. What are variables in Python? How do you declare and assign values to variables?

ANS: In Python, variables are names that you can assign to a value or an object. They serve as symbolic names that reference objects stored in memory. Here's how you declare and assign values to variables in Python:

Declaration and Assignment of Variables:

1. **Variable Declaration**:
    - Unlike some other programming languages, Python does not require explicit declaration of variables before assigning values to them. You simply assign a value to a variable, and Python automatically creates the variable.
2. **Assigning Values**:
    - You can assign values to variables using the = operator.

Q3. How do you convert one data type to another in Python?

ANS: In Python, you can convert one data type to another using type conversion functions or by using constructors of the target data type. Here are the common ways to perform type conversion:

1. Implicit Type Conversion (Type Coercion):

Python automatically converts data types in certain situations, such as when performing operations between different data types. This is known as implicit type conversion. For example:

python
Copy code
```
# Implicit conversion from int to float
result = 10 / 2  # result will be 5.0 (float)

# Implicit conversion from int to complex
z = 5 + 2j
```

2. Explicit Type Conversion (Type Casting):

Explicit type conversion involves using built-in functions or constructors to convert from one data type to another explicitly. This is particularly useful when you want to convert between incompatible types or want to ensure specific behavior.

Built-in Functions for Type Conversion:

- **int()**: Converts a number or string to an integer.

  python
  Copy code
  ```
  num_str = "10"
  num_int = int(num_str)
  ```

Q4. How do you write and execute a Python script from the command line?

ANS: **Writing a Python Script:**

1. **Open a text editor**: You can use any text editor of your choice. Popular ones include VSCode, Sublime Text, Atom, Notepad++, etc.
2. **Write your Python code**: For example, create a file named `my_script.py` and write your Python code inside it. Here's a simple example:

```
def main():

  print("Hello, World!")
```

```
if __name__ == "__main__":

    main()
```

Executing a Python Script from Command Line:

Once you've written your Python script, follow these steps to execute it from the command line:

1. **Open Command Prompt (Windows) or Terminal (Mac/Linux)**: This is where you'll run the commands to execute your script.

Q5. Given a list my_list = [1, 2, 3, 4, 5], write the code to slice the list and obtain the sub-list [2, 3].

ANS: To obtain the sublist [2, 3] from the list my_list = [1, 2, 3, 4, 5], you can use Python's list slicing feature. List slicing allows you to access a portion of a list by specifying a start index and an end index (exclusive).

Here's how you can slice my_list to get [2, 3]:

```
my_list = [1, 2, 3, 4, 5]

sub_list = my_list[1:3]

print(sub_list)
```

Explanation:

- my_list[1:3]: This slice notation starts at index 1 (which corresponds to the second element, 2) and ends just before index 3 (which corresponds to the fourth element, 4). Thus, it includes elements at indices 1 and 2, but not 3.
- The slice [1:3] returns [2, 3] from my_list.

Q6. What is a complex number in mathematics, and how is it represented in Python?

ANS: In mathematics, a complex number is a number that can be expressed in the form a+bia + bia+bi, where aaa and bbb are real numbers, and iii is the imaginary unit, which satisfies the equation i2=−1i^2 = -1i2=−1.

- aaa is called the real part of the complex number.
- bbb is called the imaginary part of the complex number.

Examples of complex numbers include 3+4i3 + 4i3+4i, −2.5−1.7i-2.5 - 1.7i−2.5−1.7i, 0+i0 + i0+i, and 4i4i4i.

Representation in Python:

Python provides built-in support for complex numbers. In Python, complex numbers are represented using the following syntax:

- Use j or J to denote the imaginary unit iii. For example, 3 + 4j.

Examples of Complex Numbers in Python:

1. **Basic Complex Number**:

   ```
   z1 = 3 + 4j
   print(z1)
   ```

2. **Complex Number with Negative Real Part**:

   ```
   z2 = -2.5 - 1.7j
   print(z2)
   ```

Q7. What is the correct way to declare a variable named age and assign the value 25 to it?
ANS: To declare a variable named age and assign the value 25 to it in Python, you simply need to use the following syntax:
age = 25

Explanation:

- **Variable Declaration**: In Python, variables are created when they are first assigned a value. There's no need for explicit declaration of variable types.
- **Assignment Operator**: The = symbol is used to assign the value 25 to the variable age.
- **Variable Name**: Choose meaningful names for variables to enhance code readability. In this case, age clearly indicates that the variable stores someone's age.

Examples of Variable Declaration and Assignment:

1. **Assigning an Integer Value**:

   ```
   age = 25
   ```

2. **Assigning a Floating-Point Value**:

   ```
   pi_value = 3.14
   ```

Assigning a String Value:

```
name = "Alice"
```

Q8. Declare a variable named price and assign the value 9.99 to it. What data type does this variable belong to?

ANS: To declare a variable named price and assign the value 9.99 to it in Python, you would write:

price = 9.99

Data Type of the Variable:

The variable price in this case belongs to the data type **float** in Python.

Explanation:

- **Float Data Type**: In Python, numbers with a decimal point are automatically interpreted as floating-point numbers (floats), regardless of whether they have an explicit decimal point. Floats represent real numbers and can include a decimal point, as well as an optional exponent (e.g., 1.23, 3.0, -2.45, etc.).
- **Assignment**: The value 9.99 is assigned to the variable price. Python recognizes 9.99 as a float due to the presence of the decimal point.

Examples of Floats in Python:

float_num1 = 3.14
float_num2 = -2.5
float_num3 = 0.0

Q9. Create a variable named name and assign your full name to it as a string. How would you print the value of this variable?
ANS: To create a variable named name and assign your full name to it as a string in Python, you can follow these steps:
name = "Your Full Name”
EXAMPLE:
If your full name is "John Doe", you would assign it like this:
name = "John Doe"

Printing the Value of the Variable:

To print the value stored in the variable name, you can use the print() function in Python:

print(name)

EXAMPLE:
Continuing with the example of "John Doe":

print(name)

Q10. Given the string "Hello, World!", extract the substring "World".
ANS: To extract the substring "World" from the string "Hello, World!" in Python, you can use string slicing.

```
my_string = "Hello, World!"
substring = my_string[7:12]
print(substring)
```

Explanation:

- **String Indexing**: In Python, strings are indexed starting from 0. So, the character 'W' in "World" is at index 7 and the character 'd' is at index 11.
- **Slicing**: Python slicing notation [start:end] extracts a portion of the string starting from index start up to, but not including, index end.
- my_string[7:12] extracts characters from index 7 (inclusive) to index 12 (exclusive), resulting in the substring "World".

Q11. Create a variable named "is_student" and assign it a boolean value indicating whether you are currently a student or not.
ANS: To create a variable named is_student and assign it a boolean value indicating whether you are currently a student or not in Python, you can follow these steps:
Example:
Let's assume you are currently a student. You would assign True to indicate that you are a student.

```
is_student = True
```

If you are not currently a student, you would assign False.

```
is_student = False
```

Explanation:

- **Boolean Data Type**: In Python, True and False are the two boolean literals that represent truth values.
- **Assignment**: You directly assign either True or False to the variable is_student based on whether the condition (currently a student or not) is true or false.

Printing the Value of the Variable:

You can use the print() function to display the value of is_student:

```
Print(is_student)
```

#Example Output:

If is_student is assigned True:

```
print(is_student)
```

If is_student is assigned False: # **print(is_student)**