ASSIGNMENT – 1

Q1. Create one variable containing following type of data:

(i) string

(ii) list

(iii) float

(iv) tuple

Answer: (i) String

my_string = "Hello, World!"

- my_string is a variable containing a string "Hello, World!".

(ii) List

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

- my_list is a variable containing a list [1, 2, 3, 4, 5].

(iii) Float

my_float = 3.14

- my_float is a variable containing a float 3.14

(iv) Tuple

my_tuple = (10, 20, 30, 40, 50)

- my_tuple is a variable containing a tuple (10, 20, 30, 40, 50)

Q2. Given are some following variables containing data:

(i) var1 = ‘ ‘

(ii) var2 = ‘[DS , ML , Python]’

(iii) var3 = [‘DS’ , ’ML’ , ‘Python’]

(iv) var4 = 1.

What will be the data type of the above given variable.

Answer: (i) var1 = ‘ ‘

- This variable contains a string with a single space character.
- Data type: str (string)

(ii) var2 = ‘[DS , ML , Python]’

- This variable contains a string that looks like a list but is enclosed in single quotes and has spaces between elements.
- Data type: str (string)

(iii) var3 = [‘DS’ , ’ML’ , ‘Python’]

- ❖ This variable contains a list of strings.
- ❖ Data type: list

(iv) var4 = 1.

- This variable contains a floating-point number.
- Data type: float

Q3. Explain the use of the following operators using an example:

(i) /

(ii) %

(iii) //

(iv) **

Answer: (i) ‘/’ (Division Operator):

- ❖ The / operator is used for division in Python. It performs division of the left operand by the right operand.

Example:

```
result = 10 / 3

print(result)
```

(ii) ‘%’ (Modulus Operator):

- ❖ The % operator returns the remainder of the division of the left operand by the right operand.

Example:

```
remainder = 10 % 3

print(remainder)
```

(iii) ‘//’ (Floor Division Operator):

- ❖ The // operator performs floor division, which means it divides the left operand by the right operand and returns the largest integer less than or equal to the quotient.

Example:

```
result = 10 // 3

print(result)
```

(iv) ‘ **’ (Exponentiation Operator):

- ❖ The ** operator raises the left operand to the power of the right operand.

Example:

```
result = 2 ** 3

print(result)
```

Q4. Create a list of length 10 of your choice containing multiple types of data. Using for loop print the element and its data type.

Answer:

```
 my_list = [10, 3.14, "Hello", True, [1, 2, 3], ('a', 'b', 'c'), {"key": "value"}, None, 5 + 2j, range(5)]

for element in my_list:

    print(f"Element: {element} \t Type: {type(element)}")
```

output:

```
Element: 10      Type: <class 'int'>

Element: 3.14    Type: <class 'float'>

Element: Hello           Type: <class 'str'>

Element: True    Type: <class 'bool'>

Element: [1, 2, 3]       Type: <class 'list'>

Element: ('a', 'b', 'c')         Type: <class 'tuple'>
```

```
Element: {'key': 'value'}      Type: <class 'dict'>
Element: None           Type: <class 'NoneType'>
Element: (5+2j)         Type: <class 'complex'>
Element: range(0, 5)    Type: <class 'range'>
```

Q5. Using a while loop, verify if the number A is purely divisible by number B and if so then how many times it can be divisible.

Answer: #CODE:

```
A = int(input("Enter number A: "))
B = int(input("Enter number B (divisor): "))
count = 0
while A % B == 0:
    A = A // B
    count += 1
print(f"{A} can be divided by {B} {count} times.")
```

Q6. Create a list containing 25 int type data. Using for loop and if-else condition print if the element is divisible by 3 or not.

Answer: #CODE:

```
my_list = [11, 9, 24, 5, 3, 18, 7, 21, 13, 30,  8, 17, 19, 22, 12, 6, 16, 4, 27, 10, 14, 25, 20, 15, 23]
for number in my_list:
    if number % 3 == 0:
        print(f"{number} is divisible by 3")
    else:
        print(f"{number} is not divisible by 3")
```

Q7. What do you understand about mutable and immutable data types? Give examples for both showing this property.

Answer: # Immutable data type:

- Immutable objects are those whose state (the data they hold) cannot be modified after they are created.
- If you want to change an immutable object, you must create a new object with the desired value.
- Examples of immutable data types in Python include int, float, bool, str, tuple, and frozenset.

# Mutable data type:

- Mutable objects are those whose state can be modified after they are created.
- Changes to mutable objects directly affect the object itself without creating a new object.
- Examples of mutable data types in Python include `list`, `dict`, `set`, and `bytearray`.