Assignment-1

Q1. Create one variable containing following type of data:

(i) string

(ii) list

(iii) float

(iv) tuple

ANS ;

(i)String ;

a = “hello world”

(ii) list;

a = [1,2,3,25.55,"hello",True]

(iii) float;

a = 25.55

(iv) tuple;

a = (1,2,3,4)

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.

ANS:

The data types of the given variables are as follows:

(i) var1: string

(ii) var2: string

(iii) var3: list

(iv) var4: float

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

(i) /

(ii) %

(iii) //

(iv) **

ANS:

(i) /

(i) The "/" operator is the division operator. It is used to divide one number by another and obtain the quotient or the result of the division. Here's an example:

a = 10

b = 3

```
c = a / b

print(c)

Output:

3.3333333333333335
```

(ii)%

The "%" operator is the modulus operator. It is used to calculate the remainder of a division operation. Here's an example:

```
a = 10

b = 3

c = a % b

print(c)

Output:

1
```

(iii)//

The "//" operator is the floor division operator. It performs division similar to the "/" operator but rounds the result down to the nearest whole number. Here's an example:

```
a = 10

b = 3

c = a // b

print(c)
```

Output:

```
3
```

(iv)**

The "**" operator is the exponentiation operator. It is used to raise a number to a certain power. Here's an example:

```
a = 10

b = 3

c = a ** b

print(c)
```

Output:

```
1000
```

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.

```
a = [1,2,3,25.55,True,"hello",55036,False,1,"pw skills",3.5,(1,2)]

for i in a:

    print(type(i))
```

Output:

```
<class 'int'>

<class 'int'>

<class 'int'>

<class 'float'>

<class 'bool'>

<class 'str'>

<class 'int'>

<class 'bool'>

<class 'int'>

<class 'str'>
```

```
<class 'float'>

<class 'tuple'>
```

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.

```
def count_divisions(a, b):

    count = 0



    while a % b == 0:

        a = a / b

        count += 1



    return count



# Example usage

number_a = 120

number_b = 2
```

```
divisions = count_divisions(number_a, number_b)

print(f"The number {number_a} is divisible by {number_b} a total of {divisions} 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.

```
numbers = [7, 12, 15, 22, 33, 40, 48, 50, 62, 75, 81, 90, 95, 100, 105, 110, 123, 130, 138, 145, 150, 165, 170, 180,
195]


for number in numbers:

    if number % 3 == 0:

        print(number, "is divisible by 3")

    else:

        print(number, "is not divisible by 3")
```

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

MUTABLE:

On the other hand, a mutable data type allows its state to be modified after it is created. This means that you can change the internal values or properties of a mutable object without creating a new object.

EXAMPLE:

```
numbers = [1, 2, 3]

numbers.append(4)    # Modifies the existing list

numbers[1] = 5       # Modifies the value at index 1
```

INNMUTABLE:

An immutable data type is one whose state cannot be modified after it is created. This means that once an object of an immutable data type is assigned a value, that value cannot be changed. Instead, any operation that appears to modify an immutable object actually creates a new object with the updated value.

Example:

String:

```
name = "Alice"

name = name + " Smith"  # This creates a new string with the concatenated value
```

[ ]: