NAME : AWANTIKA KHATRI

STUDENT ID : BIT-24S-015

DEPARTMENT : INFORMATION TECHNOLOGY

SUBJECT : ARTIFICIAL TECHNOLOGY

LAB:1

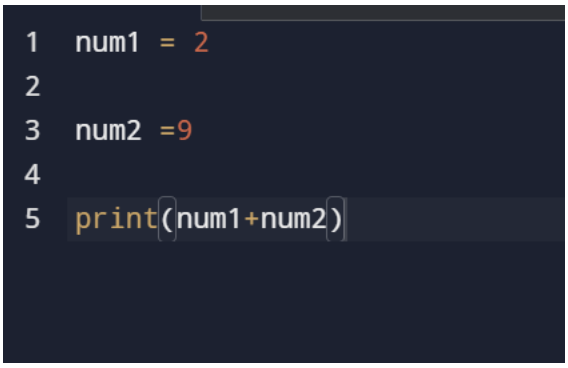
TASK:1

Make 2-2 programs of each datatype.

- **NUMERIC TYPES**

INTEGAER(INT)

#progran 1



```
num1 = 2

num2 =9

print(num1+num2)
```

OUTPUT

```
11

=== Code Execution Successful ===
```

#progran 2

```
num3=12
num4=9
print(num3+num4)
```

OUTPUT

```
21

=== Code Execution Successful ===
```

- **FLOATING-POINT(Float)**

#Program 1

```
1  pi = 3.14
2
3  radius = 5.0
4
5  print(pi * radius ** 2)
```

OUTPUT

```
78.5

=== Code Execution Successful ===
```

#Program 2

```
1  num1 = 10.5
2
3  num2 = 2.5
4
5  print(num1 / num2)
```

OUTPUT

```
4.2

=== Code Execution Successful ===
```

- **COMPLEX NUMBER**

#Program 1

```
num1 = 2 + 3j
num2 = 1 + 2j
result = num1 + num2
print("The result of addition is:", result)
```

OUTPUT

```
The result of addition is: (3+5j)

=== Code Execution Successful ===
```

 # program 2

```
num1 = 5 + 6j
num2 = 2 + 3j
result = num1 - num2
print("The result of subtraction is:", result)

```

OUTPUT

```
The result of subtraction is: (3+3j)

=== Code Execution Successful ===
```

- **Sequence Types**

String

#Program 1

```python
name = "AWANTIKA"
age = 20
print("My name is", name, "and I am", age, "years old.")

```

OUTPUT

```
My name is AWANTIKA and I am 20 years old.

=== Code Execution Successful ===
```

#Program 2

```python
text = "Hello, World!"
print(text.upper())

```

OUTPUT

```
HELLO, WORLD!

=== Code Execution Successful ===
```

- **List**

#Program 1

```
fruits = ["apple", "banana", "cherry"]
print(fruits[0])
```

OUTPUT

```
apple

=== Code Execution Successful ===
```

#Program 2

```
numbers = [1, 2, 3, 4, 5]
print(numbers[1:3])

```

OUTPUT

```
[2, 3]

=== Code Execution Successful ===
```

- **TUPLE(tuple):**

#program 1

```
coordinates = (10, 20)
values = (1, 2, 3, 4)
list_values = list(values)
print(list_values)
```

OUTPUT

```
[1, 2, 3, 4]

=== Code Execution Successful ===
```

#program 2

```
mixed_tuple = (1, "hello", 3.14)
print(mixed_tuple)
```

OUTPUT

```
(1, 'hello', 3.14)

=== Code Execution Successful ===
```

- **RANGE(range):**

#PROGRAM 1

```
r1 = range(5)
r2 = range(1, 10, 2)
range_list = list(r1)
print(range_list)
```

OUTPUT

```
[0, 1, 2, 3, 4]

=== Code Execution Successful ===
```

#program 2

```
for i in range(0, 10, 2):
    print(i)
```

OUTPUT

```
0
2
4
6
8

=== Code Execution Successful ===
```

- **SETS TYPES**

SET(sets)

#program 1

```
unique_numbers = {1, 2, 3, 3}
char_set = {'a', 'b', 'c'}
set_list = list(unique_numbers)
print(set_list)
```

OUTPUT

```
[1, 2, 3]

=== Code Execution Successful ===
```

#program 2

```
simple_set = {1, 2, 3}
simple_set.add(4)
print("Simple Set:", simple_set)
```

OUTPUT

```
Simple Set: {1, 2, 3, 4}

=== Code Execution Successful ===
```

FROZEN SETS(frozensets)

#program 1

```
frozen = frozenset([1, 2, 3])
frozen_chars = frozenset('abc')
frozen_list = list(frozen)
print(frozen_list)
```

OUTPUT

```
[1, 2, 3]

=== Code Execution Successful ===
```

#program 2

```
frozen = frozenset([1, 2, 3])
print("Frozen Set:", frozen)
```

OUTPUT

```
Frozen Set: frozenset({1, 2, 3})

=== Code Execution Successful ===
```

- **MAPPING TYPE**

DICTIONARY(dict)

#program 1

```
person = {"name": "Alice", "age": 25}
student = {"id": 101, "grade": "A"}
dict_keys = list(person.keys())
print(dict_keys)
```

OUTPUT

```
['name', 'age']

=== Code Execution Successful ===
```

#program 2

```
person = {
    "name": "Alice",
    "age": 25
}
print("Mapping Type (Dict):", person)
```

OUTPUT

```
Mapping Type (Dict): {'name': 'Alice', 'age': 25}

=== Code Execution Successful ===
```

- **BOOLEAN TYPE**

BOOLEAN(bool)

#program 1

```
is_python_fun = True
is_raining = False
bool_num = bool(1)
print(bool_num)
```

OUTPUT

```
True

=== Code Execution Successful ===
```

#program 2

```
a = 10
b = 5
result = a > b
print("Is a greater than b?", result)
```

OUTPUT

```
Is a greater than b? True

=== Code Execution Successful ===
```

TASK:2

Make up to 5 Shape programs using *.

Shape Programs

#Square

```
size = 5
for i in range(size):
    print("* " * size)
```

OUTPUT

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

=== Code Execution Successful ===
```

#Rectangle

```
width = 6
height = 4
for i in range(height):
    print("* " * width)
```

OUTPUT

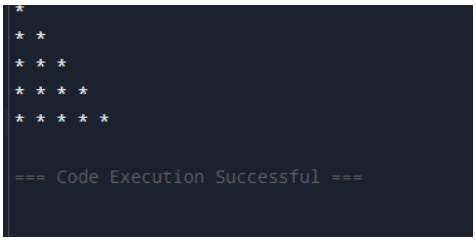
```
* * * * * *
* * * * * *
* * * * * *
* * * * * *

=== Code Execution Successful ===
```

#Triangle

```
size = 5
for i in range(size):
    print("* " * (i + 1))
```

OUTPUT



#Diamond

```
size = 5
for i in range(size):
    print(" " * (size - i - 1) + "* " * (i + 1))
for i in range(size - 2, -1, -1):
    print(" " * (size - i - 1) + "* " * (i + 1))
```

OUTPUT

#Circle

```
radius = 5
for i in range(radius):
    print(" " * (radius - i - 1) + "* " * (2 * i + 1))
for i in range(radius - 2, -1, -1):
    print(" " * (radius - i - 1) + "* " * (2 * i + 1))
```

OUTPUT

```
    *
   * * *
  * * * * *
 * * * * * * *
* * * * * * * * *
 * * * * * * *
  * * * * *
   * * *
    *

=== Code Execution Successful ===
```

TASK:3

Make same shapes you have made in task 2, using * mutiple by number.

#Square(Using Numbers)

```
size = 5
for i in range(1, size + 1):
    for j in range(1, size + 1):
        print(j, end=" ")
    print()
```

OUTPUT

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5

=== Code Execution Successful ===
```

#Rectangle(Using Numbers)

```
width = 30
height = 4

for i in range(1, height + 1):
    print(str(i) * width)
```

OUTPUT

```
1111111111111111111111111111
2222222222222222222222222222
3333333333333333333333333333
4444444444444444444444444444

=== Code Execution Successful ===
```

#TRIANGLE(Using Numbers)

```
size = 5

for i in range(1, size + 1):
    print(str(i) * i)
```

OUTPUT

```
1
22
333
4444
55555

=== Code Execution Successful ===
```

Diamond (Using Numbers)

```python
size = 5
num = 1

for i in range(size):
    print(" " * (size - i - 1) + str(num) * (i + 1))
    num += 1

for i in range(size - 2, -1, -1):
    print(" " * (size - i - 1) + str(num) * (i + 1))
    num += 1
```

OUTPUT

```
    1
   22
  333
 4444
55555
 6666
  777
   88
    9

=== Code Execution Successful ===
```

#CIRCLE(Using Numbers)

```
radius = 5
num = 1

for i in range(radius):
    print(" " * (radius - i - 1) + str(num) * (2 * i + 1))
    num += 1

for i in range(radius - 2, -1, -1):
    print(" " * (radius - i - 1) + str(num) * (2 * i + 1))
    num += 1
```

OUTPUT

```
    1
   222
  33333
 4444444
555555555
 6666666
  77777
   888
    9

=== Code Execution Successful ===
```