Saumya Deep

In [1]:

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

```
Hello World!
```

In [2]:

```
#Program to input a number from the user and display it
n = int(input("Enter a number: "))
print("Entered number is", n)
```

```
Enter a number: 74
Entered number is 74
```

In [3]:

```
#Program to add two numbers
n1 = int(input("Enter first number: "))
n2 = int(input("Enter second number: "))
print("Addition of", n1, "&", n2, "is", (n1+n2))
```

```
Enter first number: 34
Enter second number: 12
Addition of 34 & 12 is 46
```

In [4]:

```
#Program to show the use of various data types in python
# Integer (int)
num = 10
print("Integer:", num, "Type:", type(num))

# Floating point (float)
pi = 3.1415
print("Float:", pi, "Type:", type(pi))

# Complex number (complex)
comp = 2 + 3j
print("Complex Number:", comp, "Type:", type(comp))

# String (str)
text = "Hello, Python!"
print("String:", text, "Type:", type(text))

# Boolean (bool)
is_python_fun = True
print("Boolean:", is_python_fun, "Type:", type(is_python_fun))

# List (list) - Ordered, Mutable
fruits = ["Apple", "Banana", "Cherry"]
```

```
print("List:", fruits, "Type:", type(fruits))

# Tuple (tuple) - Ordered, Immutable
coordinates = (10, 20)
print("Tuple:", coordinates, "Type:", type(coordinates))

# Set (set) - Unordered, Unique elements
unique_numbers = {1, 2, 3, 4, 4, 5}
print("Set:", unique_numbers, "Type:", type(unique_numbers))

# Dictionary (dict) - Key-Value Pairs
student = {"name": "Alice", "age": 21, "course": "Data Science"}
print("Dictionary:", student, "Type:", type(student))

# NoneType (None)
nothing = None
print("NoneType:", nothing, "Type:", type(nothing))
```

```
Integer: 10 Type: <class 'int'>
Float: 3.1415 Type: <class 'float'>
Complex Number: (2+3j) Type: <class 'complex'>
String: Hello, Python! Type: <class 'str'>
Boolean: True Type: <class 'bool'>
List: ['Apple', 'Banana', 'Cherry'] Type: <class 'list'>
Tuple: (10, 20) Type: <class 'tuple'>
Set: {1, 2, 3, 4, 5} Type: <class 'set'>
Dictionary: {'name': 'Alice', 'age': 21, 'course': 'Data Science'} Type: <class 'dict'>
NoneType: None Type: <class 'NoneType'>
```

In [5]:

```
#Program to show the implementation of all the operators in python
# Arithmetic Operators
a, b = 10, 3
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Floor Division:", a // b)
print("Modulus:", a % b)
print("Exponentiation:", a ** b)

# Comparison Operators
print("Equal:", a == b)
print("Not Equal:", a != b)
print("Greater:", a > b)
print("Less:", a < b)
print("Greater or Equal:", a >= b)
print("Less or Equal:", a <= b)

# Logical Operators
x, y = True, False
print("AND:", x and y)
print("OR:", x or y)
print("NOT:", not x)

# Bitwise Operators
m, n = 5, 3
print("Bitwise AND:", m & n)
print("Bitwise OR:", m | n)
print("Bitwise XOR:", m ^ n)
print("Bitwise NOT:", ~m)
print("Left Shift:", m << 1)
print("Right Shift:", m >> 1)

# Assignment Operators
x = 5
x += 3
print("x after +=3:", x)
x *= 2
print("x after *=2:", x)

# Identity Operators
list1 = [1, 2, 3]
list2 = list1
list3 = [1, 2, 3]
print("list1 is list2:", list1 is list2)
print("list1 is not list3:", list1 is not list3)

# Membership Operators
print("2 in list1:", 2 in list1)
print("5 not in list1:", 5 not in list1)

# Ternary Operator
a, b = 15, 25
min_value = a if a < b else b
print("Minimum Value:", min_value)
```

```
Addition: 13
Subtraction: 7
Multiplication: 30
Division: 3.3333333333333335
Floor Division: 3
Modulus: 1
Exponentiation: 1000
Equal: False
Not Equal: True
Greater: True
Less: False
Greater or Equal: True
Less or Equal: False
AND: False
OR: True
NOT: False
Bitwise AND: 1
Bitwise OR: 7
Bitwise XOR: 6
Bitwise NOT: -6
Left Shift: 10
Right Shift: 2
x after +=3: 8
x after *=2: 16
list1 is list2: True
list1 is not list3: True
2 in list1: True
5 not in list1: True
Minimum Value: 15
```