University Of Loralai

Name: Sajad ullah .

Semester 3nd semester .

Dept Computer Science .

Course Data Engineering .

Course instructor Sir.Hazrat_Bilal .

Lecture no 2nd Lecture .

1. nput and Output in Python

- `input()` → takes data from user
- `print()` → shows output on screen

```
name = input("Enter your name: ")
print("Welcome", name)
```

With numbers:

```
age = int(input("Enter your age: "))
print("Your age is:", age)
```

---

2. Variables in Python

A variable stores data in memory.

```
name = "Ali"
age = 20
marks = 85.5

print(name)
print(age)
print(marks)
```

Rules:

- Start with letter or _
- No spaces
- Case-sensitive (`Age` ≠ `age`)

---

3. Comments and Its Types

Comments are used to explain code.

- **Single-line comment**

  ```
  # This is a comment
  print("Hello")
  ```

- **Multi-line comment**

  ```
  """
  This is
  a multi-line
  ```

```
        comment
        """
        print("Python")
```

4. Data Types (Two Types: Numeric & Non-Numeric)

Numeric:

- `int` → 10
- `float` → 3.5
- `complex` → 2+3j

```
a = 10
b = 3.5
c = 2 + 3j

print(type(a))
print(type(b))
print(type(c))
```

Non-Numeric (basic):

- `str` → "Hello"
- `list` → [1,2,3]
- `bool` → True/False

5. Relational Operations (Comparison)

Used to compare two values. Result is **True/False**.

Operator	Meaning
>	Greater
<	Less
>=	Greater or equal
<=	Less or equal
==	Equal
!=	Not equal

```
a = 10
b = 5

print(a > b)     # True
print(a == b)    # False
print(a != b)    # True
```

6. Logical Operations

Used with conditions.

Operator	Meaning
And	Both true
Or	Any one true
Not	Reverse result

```
a = 10
b = 5

print(a > 5 and b < 10)    # True
print(a < 5 or b < 10)     # True
print(not(a > b))          # False
```

7. Bitwise Operations

Work on binary numbers.

Operator	Name
&	AND
`	`
^	XOR
~	NOT
<<	Left shift
>>	Right shift

```
a = 5   # 101
b = 3   # 011

print(a & b)    # 1
print(a | b)    # 7
print(a ^ b)    # 6
print(a << 1)   # 10
print(a >> 1)   # 2
```

8. Lists in Python

List stores multiple values.

```
marks = [70, 80, 90, 85]
print(marks)
print(marks[0])    # First item
```

Change value:

```
marks[1] = 95
print(marks)
```

Add item:

```
marks.append(88)
print(marks)
```

---

9. Strings and Its Types

String = text inside quotes.

```
name = "Sajad"
city = 'Ziarat'
```

- **Types (Ways to write strings)**

```
s1 = "Hello"
s2 = 'World'
s3 = """This is
multi-line
string"""
```

String operations:

```
text = "Sajad"

print(len(text))      # Length
print(text.upper())   # PYTHON
print(text.lower())   # python
print(text[0])        # P
```