Ch-7 Data Handling

Operators in Python

An **operator** is a symbol that performs operations on variables and values.

1. Arithmetic Operators

Operator	Description	Example
+	Addition	5 + 3 = 8
-	Subtraction	5 - 3 = 2
*	Multiplication	5 * 3 = 15
/	Division (float)	5 / 2 = 2.5
//	Floor Division	5 // 2 = 2
%	Modulus	5 % 2 = 1
**	Exponentiation	2 ** 3 = 8

2. Relational (Comparison) Operators

Operator	Description	Example
==	Equal to	5 == 5 → True
!=	Not equal to	5 != 3 → True
>	Greater than	5 > 3 → True
<	Less than	5 < 3 → False
>=	Greater or equal	5 >= 5 → True
<=	Less or equal	5 <= 4 → False

3. Logical Operators

Operator	Description	Example
and	True if both conditions are true	(5 > 3) and (3 > 1) → True
or	True if any one is true	(5 > 3) or (3 < 1) → True
not	Reverses result	not(5 > 3) → False

4. Assignment Operators

Operator	Description	Example
=	Assigns value	x = 5
+=	Add and assign	x += 3 → x = x + 3
-=	Subtract and assign	x -= 1
*=	Multiply and assign	x *= 2
/=	Divide and assign	x /= 2
%=	Modulus and assign	x %= 2

5. Identity Operators

Operator	Description	Example
is	True if both refer to same object	x is y
is not	True if not the same object	x is not y

```
x = [1, 2]
y = x
print(x is y)  # True (same memory)
```

6. Membership Operators

Operator	Description	Example
in	True if value exists	'a' in 'apple' → True
not in	True if value does not exist	'z' not in 'apple'→ True

Expressions, Statements & Type Conversion

Expression

An expression is a combination of variables, constants, and operators that evaluates to a value.

```
x = 10 + 5 * 2   # expression: 10 + 5 * 2
```

Statement

A line of code that performs an action. It may contain expressions.

```
x = 20  # assignment statement
print(x)  # output statement
```

Precedence of Operators

Precedence Level	Operators
Highest	**
Next	*, /, //, %
Next	+, -
Next	Relational: ==, !=, >, <, >=, <=
Lowest	Logical: not, and, or

Use **parentheses ( )** to override default precedence.

Type Conversion

Implicit Conversion

Automatically handled by Python:

```
x = 10 + 3.5  # x becomes 13.5 (int + float → float)
```

Explicit Conversion (Type Casting)

Function	Converts to
int()	Integer
float()	Float
str()	String

```
x = int("5")     # 5
y = float(10)    # 10.0
```

Input and Output Functions

input() – Accept user input

```
name = input("Enter your name: ")
```

Note: input() returns a **string**, use int() or float() for numeric input.

print() – Display output

```
print("Welcome", name)
```

Errors in Python

Type	Description	Example
Syntax Error	Violation of language rules (code won't run)	print("Hello' → missing quote
Logical Error	Code runs but gives wrong output	Incorrect formula
Runtime Error	Error during execution (e.g., divide by zero)	x = 10 / 0 → ZeroDivisionError