

# Table of contents

Press Escape key to exit fullscreen.

## 01

### Operator

Types of Operators

## 02

### Operator Precedence

Operator Precedence

## 03

### Type Conversion

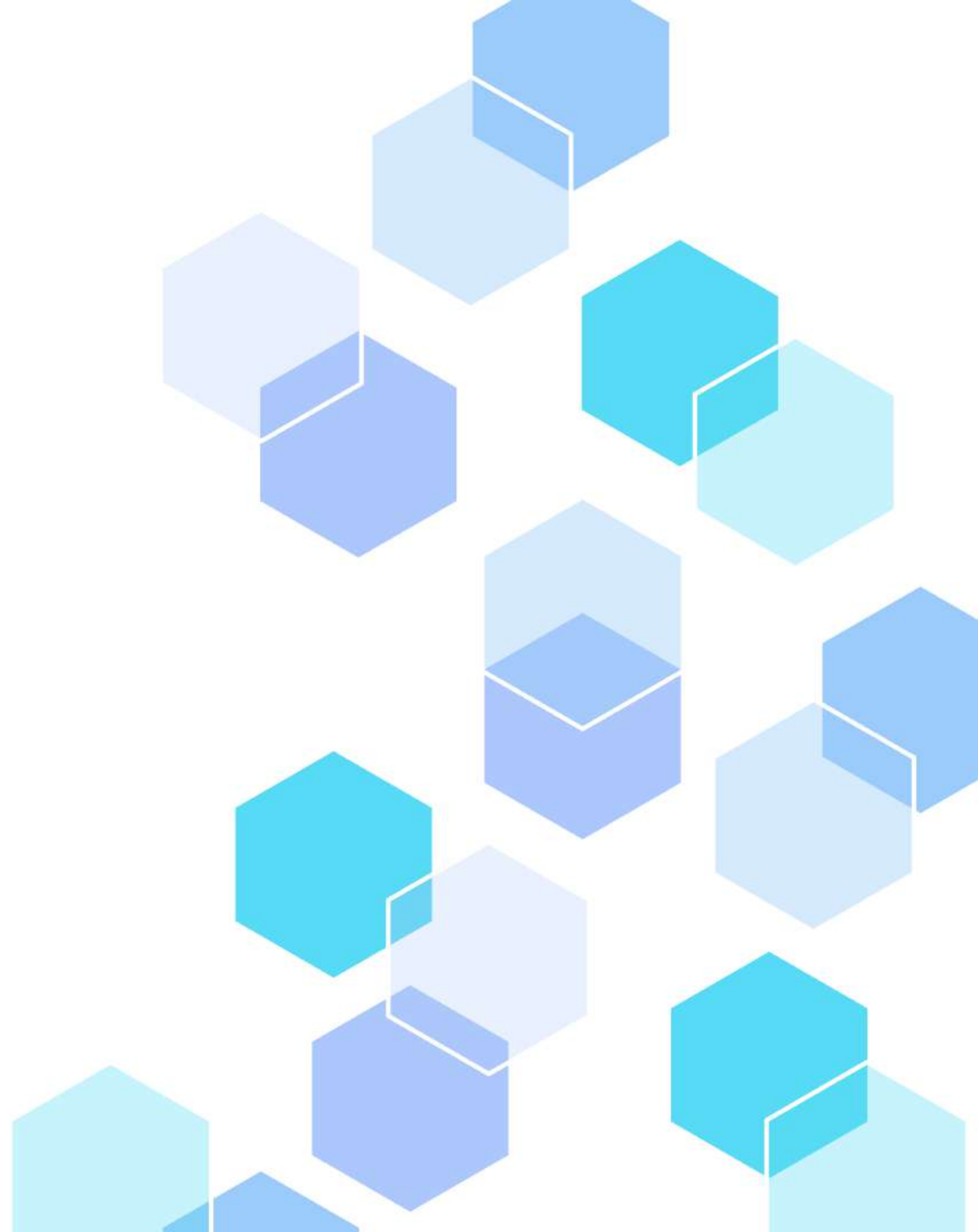
Implicit Type Conversion,  
Explicit Type Conversion

---

# 01

# Operator

Intro about DSA



# Operator

An operator is a **symbol that represents a specific operation** on one or more operands.

Operators are fundamental to programming languages as they enable you to perform various computations and manipulate data.





# Types of Operators

## Arithmetic Operators

+, -, \*, /, %

## Relational Operators

<, <=, >, >=, ==, !=

## Logical Operators

&&, ||, !

## Assignment Operators

=, +=, -=, \*=, /=, %=

## Miscellaneous Operators

?:, sizeof, \*, &

## Unary Operators

++, --



---

# 02

# Operator Precedence

Use case scenario



# Operator Precedence

---

**Operator precedence** in C **determines** the **order** in which operators are evaluated when an expression contains multiple operators.

Operators with higher precedence are evaluated before operators with lower precedence.



# Operator Precedence

Type of Operator	Associativity	Category
() ++ --	Left to right	Postfix
(type)* & sizeof	Right to left	The Unary Operator
/ * %	Left to right	The Multiplicative Operator
- +	Left to right	The Additive Operator
< > >= <=	Left to right	The Relational Operator
!= ==	Left to right	The Equality Operator
&&	Left to right	Logical AND
	Left to right	Logical OR
?:	Right to left	Conditional
= -= += /= *= %=	Right to left	Assignment
,	Left to right	Comma

---

# 03

# Type Conversion

Implicit Type Conversion,  
Explicit Type Conversion





# Type Conversion

**Type conversion** in C++ language, also known as **type casting**, refers to the **process of converting** a value from one data type to another.

# Types of Type Conversion

**Implicit Type  
Conversion  
(Type Coercion)**

**Explicit Type  
Conversion  
(Type Casting)**

# Implicit Type Conversion

Implicit type conversion is **performed by the compiler** automatically during compilation.

```
int num1 = 10;  
float num2 = 5.5;
```

```
float ans = num1 + num2;
```

```
// num1 is implicitly converted to float before addition
```



# Explicit Type Conversion

Explicit type conversion, or **type casting**, is **done by the programmer explicitly using casting operators**.

Type casting is performed using casting operators like **(type)**

```
int num1 = 10;  
float num2 = 5.5;  
  
int ans = num1 + (int)num2;  
// num2 is explicitly cast to int before addition
```

# TL;DR

## Operator

A symbol that represents a specific operation

Types of Operators

Operator Precedence

## Type Conversion

The process of converting a value from one data type to another.

Implicit Conversion

Explicit Conversion