**Lecture 8 - C++ Operators**

**1. Basic Concepts**

**Operator**

* A symbol that performs an operation
* Examples: +, -, \*, /, =, ==

**Operands**

* Values on which operators work
* Example: In 10 + 5, both 10 and 5 are operands

**Expression**

* Valid combination of operands + operators that produces a result
* Example: a + b \* 2

**Valid vs Invalid Expressions:**

✅ **Valid:**

* a + b;
* 5 \* (2 + 3);
* x = y - 7;

❌ **Invalid:**

* +; (operator without operands)
* a +; (missing second operand)
* = 10; (missing variable)

**Statement**

* Complete instruction that performs some action
* Always ends with semicolon (;)
* Example: int a = 5;

**Key Differences**

**Expression vs Statement:**

* **Expression:** Produces a value (e.g., a + b)
* **Statement:** Performs action, ends with ; (e.g., x = a + b;)

**2. Types of Operators**

**A. Arithmetic Operators**

| **Operator** | **Name** | **Function** |
| --- | --- | --- |
| + | Addition | Adds two numbers |
| - | Subtraction | Subtracts second from first |
| \* | Multiplication | Multiplies two numbers |
| / | Division | Divides first by second |
| % | Modulus | Returns remainder of division |

**Division Rules:**

* int / int → Result is integer (truncated)
* float / int → Result is float
* double / int → Result is double

**Modulus (%) Operator:**

* Used to find remainder
* Formula: a % b gives remainder when a is divided by b
* Useful for checking divisibility

**B. Relational Operators**

| **Operator** | **Meaning** | **Result** |
| --- | --- | --- |
| == | Equal to | true (1) or false (0) |
| != | Not equal to | true (1) or false (0) |
| > | Greater than | true (1) or false (0) |
| < | Less than | true (1) or false (0) |
| >= | Greater than or equal | true (1) or false (0) |
| <= | Less than or equal | true (1) or false (0) |
|  |  |  |

**C. Logical Operators**

| **Operator** | **Name** | **Function** |
| --- | --- | --- |
| && | Logical AND | True only if both conditions are true |
| || | Logical OR | True if at least one condition is true |
| ! | Logical NOT | Reverses the boolean value |

**D. Assignment Operators**

| **Operator** | **Meaning** | **Equivalent** |
| --- | --- | --- |
| = | Assignment | Assign value |
| += | Add and assign | a = a + value |
| -= | Subtract and assign | a = a - value |
| \*= | Multiply and assign | a = a \* value |
| /= | Divide and assign | a = a / value |
| %= | Modulus and assign | a = a % value |

**E. Increment and Decrement Operators**

**Pre-increment/decrement:**

* ++a: Increment first, then use
* --a: Decrement first, then use

**Post-increment/decrement:**

* a++: Use first, then increment
* a--: Use first, then decrement

**3. Operator Precedence (Priority Order)**

1. **Parentheses** ()
2. **Increment/Decrement** ++, --
3. **Multiplication, Division, Modulus** \*, /, % (Left to Right)
4. **Addition, Subtraction** +, - (Left to Right)
5. **Relational** <, >, <=, >=
6. **Equality** ==, !=
7. **Logical AND** &&
8. **Logical OR** ||
9. **Assignment** =, +=, -=, etc.

**Key Points:**

* Same precedence operators are evaluated left to right
* Use parentheses to override default precedence
* In programming: left-to-right evaluation for same precedence

**4. Sample Programs**

**Program 1: Rectangle Perimeter**

#include <iostream>

using namespace std;

int main() {

float length = 10, breadth = 5;

float perimeter = 2 \* (length + breadth);

cout << "Perimeter = " << perimeter << endl;

return 0;

}

**Program 2: Celsius to Fahrenheit**

#include <iostream>

using namespace std;

int main() {

float celsius;

cout << "Enter temperature in Celsius: ";

cin >> celsius;

float fahrenheit = (celsius \* 9/5) + 32;

cout << "Temperature in Fahrenheit = " << fahrenheit << endl;

return 0;

}

**5. Practice Questions & Solutions**

**Question 1: Basic Precedence**

int a = 2 + 3 \* 4;

cout << a << endl;

**Answer:** 14 (because \* has higher precedence than +)

**Question 2: Post-increment**

int x = 10;

int y = x++ + 5;

cout << y;

**Answer:** 15 (x is used first as 10, then incremented)

**Question 3: Logical AND**

int a = 5, b = 3;

int result = a > b && b < 2;

cout << result;

**Answer:** 0 (5 > 3 is true, but 3 < 2 is false; true && false = false)

**6. Important Tips**

**Division Gotchas:**

* Integer division truncates: 3/5 = 0 (not 0.6)
* Mixed types: 2.0/5 = 0.4
* Store in appropriate data type for decimal results

**Modulus Applications:**

* Check if number is even: n % 2 == 0
* Check divisibility: n % divisor == 0
* Get last digit: n % 10

**Pre vs Post Increment:**

* **Pre (++a):** Change first, then use
* **Post (a++):** Use first, then change

**Programming vs Mathematics:**

* Programming follows left-to-right for same precedence
* Mathematics might have different conventions
* Always use parentheses when in doubt

**7. Memory Tips**

* **BODMAS/PEMDAS** for precedence (with programming modifications)
* **AND (&&)** needs ALL conditions true
* **OR (||)** needs ANY condition true
* **Assignment** always goes right-to-left
* **Comparison** operators return 0 or 1 (false or true)