RELATIONAL OPERATORS

INTRODUCTION TO CONDITIONAL STATEMENT

prepared by:

Gyro A. Madrona
Electronics Engineer

TOPIC OUTLINE

Equal to Operator

Not Equal to Operator

Greater than Operator

Less than Operator

Greater than or Equal to Operator

Less than or Equal to Operator

RELATIONAL OPERATORS

RELATIONAL OPERATORS

Relational operators are used to **compare** two values or expressions. They evaluate the relationship between the operands and return a boolean value **(`true` or `false`)**. These operators are commonly used in decision-making statements like **`if`**, **`while`**, and **`for`** loops.

EQUAL TO OPERATOR

Equal to **(==)** operator evaluates if two operands are equal. It returns **true** if the values are **equal**, and **false** otherwise.

Examples:

```
int a = 5;

int b = 5;

a == b evaluates to true
```

```
int a = 0;

int b = 5;

a == b evaluates to false
```

NOT EQUAL TO OPERATOR

Not equal to `(!=)` operator evaluates if two operands are not equal. It returns **`true`** if the values are **not equal**, and **`false`** otherwise.

Examples:

```
int a = 5;

int b = 5;

a != b evaluates to false
```

```
int a = 0;

int b = 5;

a != b evaluates to true
```

GREATER THAN OPERATOR

Greater than **(>)** operator returns **true** if the left operand is **greater** than the right operand, and **false** otherwise.

Examples:

```
int a = 5;

int b = 0;

a > b evaluates to true
```

```
int a = 0;

int b = 5;

a > b evaluates to false
```

LESS THAN OPERATOR

Greater than **(<)** operator returns **true** if the left operand is **less** than the right operand, and **false** otherwise.

Examples:

```
int a = 5;

int b = 0;

a < b evaluates to false
```

```
int a = 0;

int b = 5;

a < b evaluates to true
```

GREATER THAN OR EQUAL OPERATOR

Greater than or equal **(>=)** returns **true** if the left operand is **greater than or equal** to the right operand, and **false** otherwise.

Examples:

```
int a = 3;

int b = 3;

a >= b evaluates to true
```

```
int a = 5;

int b = 3;

a >= b evaluates to true
```

LESS THAN OR EQUAL OPERATOR

Less than or equal **(<=)** returns **`true`** if the left operand is **less than or equal** to the right operand, and **`false`** otherwise.

Examples:

```
int a = 3;

int b = 3;

a <= b evaluates to true
```

```
int a = 5;

int b = 3;

a <= b evaluates to false
```

EXERCISE

```
int main() {
    int a = 10, b = 20;
    cout << "a == b: " << (a == b);
    cout << "a != b: " << (a != b);
    cout << "a > b: " << (a > b);
    cout << "a < b: " << (a < b);
    cout << "a >= b: " << (a >= b);
    cout << "a <= b: " << (a <= b);
    return 0;
}
```

LABORATORY