OPERATORS IN C

ABHISHEK P

INTRODUCTION

- Operators are fundamental building blocks in C programming that perform operations on variables and values.
- Understanding operators is essential for writing efficient and effective C code.
- The data items on which the operators are applied are known as operands.
- Operators are applied between the operands.
- Operators can be classified into :
  - Unary Operators
  - Binary Operators

UNARY OPERATORS

- Unary operators are operators that operate on a single operand.
- They are used to perform operations such as
  - Increment operator (++)
  - Decrementing operator (--)
  - Sizeof
  - (type)*

BINARY OPERATORS

- Binary operators in C operate on two operands.
- They are used to perform operations such as :
    - Arithmetic Operators
    - Relational Operators
    - Logical Operators
    - Assignment Operators
    - Bitwise Operators
    - Conditional Operators

Arithmetic Operators

- Arithmetic operators in C are used to perform basic mathematical operations on variables and values.

Arithmetic Operators

Operator	Description	Example
+	Adds two operands	A + B = 30
-	Subtracts second operand from the first	A − B = -10
*	Multiplies both operands	A * B = 200
/	Divides numerator by de-numerator	B / A = 2
%	Modulus Operator and remainder of after an integer division	B % A = 0

Relational Operators

- Relational operators in C are used to compare two values or expressions.
- They return a boolean result, which is represented as 1 (true) or `0` (false).

Relational Operators

Operator	Description	Example
==	Checks if two operands are equal.	(A == B)
!=	Checks if two operands are not equal	(A != B)
>	Checks if the first operand is greater than the second.	(A > B)
<	Checks if the first operand is less than the second.	(A < B)
>=	Checks if the first operand is greater than or equal to the second.	(A >= B)
<=	Checks if the first operand is less than or equal to the second.	(A <= B)

Logical Operators

- Logical operators are used to perform logical operations on boolean values, which are typically `true` or `false`.
- These operators are fundamental in programming for controlling the flow of programs and making decisions based on multiple conditions.

Logical Operators

Operator	Description	Example
&&	The AND operator returns true if both operands are true. Otherwise it returns false.	(A && B)
||	The OR operator returns true if at least one of the operands is true If both are false, it returns false.	(A || B)
!	The NOT operator inverts the boolean value. If the value is true, it returns false, and if it’s false, it returns true.	!(A && B)

Assignment Operator

- Assignment operators are a fundamental concept in programming that allow you to assign values to variables.
- They come in various forms, with the most basic being the simple assignment operator (=).

Assignment Operator

operator	Description	Example
=	Simple assignment operator. To Assigns values	C = A + B
+=	Add AND assignment operator. It adds the right operand to the left operand and assign the result to the left operand.	(C += A) / (C=C+A)
-=	Subtract AND assignment operator. It subtracts the right operand from the left operand and assigns the result to the left operand.	(C -= A) / (C = C - A)
*=	Multiply AND assignment operator. It multiplies the right operand with the left operand and assigns the result to the left operand	(C *= A) / (C = C * A)

Assignment Operator

operator	Description	Example
/=	Divide AND assignment operator. It divides the left operand with the right operand and assigns the result to the left operand.	(C /= A) / (C = C / A)
%=	Modulus AND assignment operator. It takes modulus using two operands and assigns the result to the left operand.	(C %= A) / (C = C % A)
<<=	Left shift AND assignment operator.	(C <<= 2) / (C = C << 2)
>>=	Right shift AND assignment operator	(C >>= 2) / (C = C >> 2)

Assignment Operator

operator	Description	Example
&=	Bitwise AND assignment operator.	(C &= 2) / (C = C & 2)
^=	Bitwise exclusive OR and assignment operator.	(C ^= 2) / (C = C ^ 2)
|=	Bitwise inclusive OR and assignment operator	(C |= 2) /(s C = C | 2)

Bitwise Operator

- The bitwise operators are the operators used to perform the operations on the data at the bit-level.
- It is also known as bit-level programming.
- It consists of two digits, either 0 or 1.

Bitwise Operator

Operator	Meaning of operator	example
&	Bitwise AND operator	A&B
|	Bitwise OR operator	A|B
^	Bitwise exclusive OR operator	A^B
~	One's complement operator (unary operator)	~A
<<	Left shift operator	A<<B
>>	Right shift operator	A>>B

Conditional Operator

- The conditional operator is also known as a ternary operator.
- The conditional statements are the decision-making statements
- The behavior of the conditional operator is similar to the 'if-else
- Syntax -
  - Expression1? expression2: expression3;

Expression 1? expression 2: expression 3

- the expression1 is a Boolean condition that can be either true or false value.
- If the expression 1 results into a true value, then the expression 2 will execute.
- The expression 2 is said to be true only when it returns a non-zero value.
- If the expression 1 returns false value then the expression 3 will execute.
- The expression 3 is said to be false only when it returns zero value.

Example

```
#include

 int main(){

int age; // variable declaration

printf("Enter your age");

 scanf("%d",&age); // taking user input for age variable

 (age>=18)? (printf("eligible for voting")) : (printf("not eligible for voting")); // conditional operator

 return 0;

 }
```