PROGRAMMING FOR PROBLEM SOLVING

UNIT 2 ONE SHOT

WELCOME

INTELLIGENCE LEARNING

UNIT 2 BCS101 / BCS201: PROGRAMMING FOR PROBLEM SOLVING

Ques. What is an operator ? List all the operator in c with example.

Ans. Operators are special symbols whose meaning is already know to C complier. There are 45 operators in C classified as:-

1. **Unary Operators:-** Operators that need only one operand to complete its task is termed as Unary operator.

2. **Binary Operator:-** Operator that need two operand to complete its task is termed as Binary operators.

3. **Ternary Operator:-** Operators that need three operand to perform its task is termed as Ternary operator or conditional operator.

General Categories of Operator

1. Arithmetic Operators

Used for basic mathematical operations.

Operator	Meaning	Example
+	Addition	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	Division	a / b
%	Modulus	a % b

2. Relational (Comparison) Operators

Used to compare values.

Operator	Meaning	Example
==	Equal to	a == b
!=	Not equal to	a != b
>	Greater than	a > b
<	Less than	a < b
>=	Greater or equal	a >= b
<=	Less or equal	a <= b

3. Logical Operators

Used for logical conditions.

Operator	Meaning	Example
&&	Logical AND	(a > 5 && b < 10)
||	Logical OR	(a > 5 || b < 10)
!	Logical NOT	!(a > 5)

Truth Table for Logical Operators in C

1. Logical AND (&&)

A (Condition 1)	B (Condition 2)	A && B (Result)
0 (false)	0 (false)	0 (false)
0 (false)	1 (true)	0 (false)
1 (true)	0 (false)	0 (false)
1 (true)	1 (true)	1 (true)

Truth Table for Logical Operators in C

2. Logical OR (||)

A (Condition 1)	B (Condition 2)	A || B (Result)
0 (false)	0 (false)	0 (false)
0 (false)	1 (true)	1 (true)
1 (true)	0 (false)	1 (true)
1 (true)	1 (true)	1 (true)

Truth Table for Logical Operators in C

3. Logical NOT (!)

A (Condition)	!A (Result)
0 (false)	1 (true)
1 (true)	0 (false)

4. Assignment Operators

Used to assign values to variables.

Operator	Meaning	Example
=	Assign	a = 10
+=	Add and assign	a += 5
-=	Subtract and assign	a -= 5
*=	Multiply and assign	a *= 2
/=	Divide and assign	a /= 2
%=	Modulus and assign	a %= 3

5. Increment and Decrement Operators

Operator	Meaning	Example
++	Increment by 1	a++ or ++a
--	Decrement by 1	a-- or --a

6. Bitwise Operators

Works on bits and performs bit-by-bit operation.

Operator	Meaning	Description
&	Bitwise AND	Sets bit to 1 if both bits are 1
|	Bitwise OR	Sets bit to 0 if both bits are 0
^	Bitwise XOR	Sets bit to 1 if only one bit is 1
~	Bitwise NOT	Flips all bits
<<	Left Shift	Shifts bits to the left
>>	Right Shift	Shifts bits to the right

Truth Table for Bitwise Operators in C

Operator	Description	Expression
&	Bitwise AND	A & B
|	|	Bitwise OR
^	Bitwise XOR	A ^ B
~A	Bitwise NOT (1's comp)	~A
<<	Left Shift	A << 1
>>	Right Shift	A >> 1

A (bit)	B (bit)	A & B	A | B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

C program to find the largest of three numbers using the ternary operator

```
#include <stdio.h>
int main() {
    int a, b, c, largest;

    // Input three numbers
    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    // Using ternary operator to find the largest
    largest = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

    // Display the result
    printf("The largest number is: %d\n", largest);
    return 0;
}
```

Difference Between Precedence and Associativity in C

Point	Precedence	Associativity
Meaning	Tells which operator works first	Tells which side to start from
Use	Used when there are different operators	Used when there are same level operators
Example	a + b * c → * is done first	a = b = c → right side is done first
Direction	Not about direction	It decides left-to-right or right-to-left
Helps to	Solve priority of operations	Solve tie when priority is same
Type	Every operator has its own level	Some are left-to-right, some right-to-left

What is Type Conversion in C? Difference Between Implicit and Explicit Type Conversion.

Type conversion means changing a value from one data type to another.

Example: Converting an int to float, or char to int.

Feature	Implicit Type (Conversion)	Explicit Type Conversion (Type Casting)
Who does it?	Done by the compiler	Done by the programmer
Syntax	Automatic – no extra code needed	Use casting like (data_type) before value
Also called	Type Conversion	Type Casting
Risk of Data Loss?	Less (safe conversions only)	Yes (you may lose data if cast to smaller type)
Example	int a = 10; float b = a; → a changes to float	float a = 5.7; int b = (int)a; → b = 5

Write short note on various decision / selection control instructions use in C ?

These are used to make decisions based on conditions (true or false). They help the program choose different actions.

1. if Statement
2. if-else Statement
3. else-if Ladder
4. Nested if
5. Switch Statement

1 if Statement

Checks a condition. Executes the block only if the condition is **true**.

Syntax:

```
if (condition) {
    // code to execute
}
```

2 if-else Statement

Executes one block if the condition is true, another block if false.

Syntax:

```
if (condition) {
    // true block
} else {
    // false block
}
```

3 else-if Ladder

Used to check multiple conditions in sequence. Executes the first true block.

Syntax:

```
if (condition1) {
    // block 1
} else if (condition2) {
    // block 2
} else {
    // default block
}
```

4 Nested if

if statement inside another if. Used for multiple-level decision making.

Syntax:

```
if (condition1) {
    if (condition2) {
        // code block
    }
}
```

5 Switch Statement

Best for multi-way branching. Used when a variable is compared with fixed values.

Syntax:

```
switch (variable) {
    case value1:
        // code
        break;
    case value2:
        // code
        break;
    default:
        // default code
}
```

C Program to Check Leap Year

```
#include <stdio.h>
int main() {
    int year;

    printf("Enter a year: ");
    scanf("%d", &year);

    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
        printf("%d is a leap year.\n", year);
    } else {
        printf("%d is not a leap year.\n", year);
    }
    return 0;
}
```

C Program to Find the Largest of Three Numbers

```
#include <stdio.h>
int main() {
    int a, b, c;
    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    if (a >= b && a >= c) {
        printf("%d is the largest number.\n", a);
    } else if (b >= a && b >= c) {
        printf("%d is the largest number.\n", b);
    } else {
        printf("%d is the largest number.\n", c);
    }
    return 0;
}
```

Ques. Write a program to check if a student is eligible for admission based on:

Math ≥ 60

Physics ≥ 50

Chemistry ≥ 40

AND either:

Total marks in all three subjects ≥ 200

Or total in Math + Physics ≥ 150

```
#include <stdio.h>
#include <conio.h>

void main() {
    int m, p, c, total, mp_total;

    clrscr();  // Clear screen (Turbo C specific, optional)

    printf("Enter marks of Math, Physics, and Chemistry:\n");
    scanf("%d %d %d", &m, &p, &c);

    total = m + p + c;
    mp_total = m + p;
```

```
if (m >= 60 && p >= 50 && c >= 40) {
        if (total >= 200 || mp_total >= 150) {
            printf("Eligible\n");
        } else {
            printf("Not Eligible\n");
        }
    } else {
        printf("Not Eligible\n");
    }

    getch();  // Wait for key press (Turbo C specific)
}
```

C Program: Check Character Type
Uppercase letter
Lowercase letter
Digit (numeric)
Symbol

```
#include <stdio.h>
#include <conio.h>

void main() {
   char ch;

   clrscr();  // Clear screen (Turbo C specific)

   printf("Enter a character:\n");
   scanf("%c", &ch);
```

```
if (ch >= 65 && ch <= 90) {
        printf("Uppercase character\n");
    } else if (ch >= 97 && ch <= 122) {
        printf("Lowercase character\n");
    } else if (ch >= 48 && ch <= 57) {
        printf("Numeric character\n");
    } else {
        printf("Symbol\n");
    }

    getch();  // Wait for key press (Turbo C specific)
}
```

C Program to Find Quadrant Using if-else Ladder.

```
#include <stdio.h>

int main() {
    int x, y;

    // Input coordinates
    printf("Enter X and Y coordinates:\n");
    scanf("%d %d", &x, &y);
```

```
// Determine the quadrant
    if (x > 0 && y > 0) {
        printf("Point lies in First Quadrant\n");
    } else if (x < 0 && y > 0) {
        printf("Point lies in Second Quadrant\n");
    } else if (x < 0 && y < 0) {
        printf("Point lies in Third Quadrant\n");
    } else if (x > 0 && y < 0) {
        printf("Point lies in Fourth Quadrant\n");
    } else if (x == 0 && y == 0) {
        printf("Point is at Origin\n");
    } else if (x == 0) {
        printf("Point lies on Y-axis\n");
    } else if (y == 0) {
        printf("Point lies on X-axis\n");
    }

    return 0;
}
```

C program for a basic calculator using a switch statement

```
#include <stdio.h>
#include <conio.h>

void main() {
    int a, b, c, ch;
    clrscr();  // Clears the screen (specific to some compilers like Turbo C)
    printf("\n Enter 2 Numbers :\n");
    scanf("%d %d", &a, &b);

    printf("\n Enter 1 for addition : \n");
    printf(" Enter 2 for subtraction : \n");
    printf(" Enter 3 for multiplication : \n");
    printf(" Enter 4 for division : \n");

    printf("\n Now Enter your choice : \n");
    scanf("%d", &ch);
```

```
switch (ch) {
    case 1:
        c = a + b;
        printf("\n Sum is %d", c);
        break;
    case 2:
        c = a - b;
        printf("\n Sub is %d", c);
        break;
    case 3:
        c = a * b;
        printf("\n Mul is %d", c);
        break;
```

```
 case 4:
          if (b != 0) {
             c = a / b;
             printf("\n Div is %d", c);
          } else {
             printf("\n Division by zero is not allowed.");
          }
          break;
default:
          printf("\n Wrong input");
    }
    getch();
}
```

THANK YOU FOR WATCHING

DO LIKE SHARE COMMENT AND SUBSCRIBE