C-OPERATORS

1Q.What are the operators with code examples in 'c' language?

A.

In C programming, operators are special symbols that perform operations on variables and values. Here's a breakdown of common operator types with code examples:

1. Arithmetic Operators:

☑ These perform mathematical calculations.

`+` (Addition): Adds two operands.

`-` (Subtraction): Subtracts the second operand from the first.

` ` (Multiplication): Multiplies two operands.

`/` (Division): Divides the first operand by the second.

`%` (Modulo): Returns the remainder of an integer division.

```c

# include

int main() {

int a = 10, b = 3;

printf("a + b = %d\n", a + b); // Output: 13

printf("a - b = %d\n", a - b); // Output: 7

printf("a  b = %d\n", a  b); // Output: 30
```

```
printf("a / b = %d\n", a / b); // Output: 3 (integer division)

printf("a %% b = %d\n", a % b); // Output: 1

return 0;

}
```

2. Relational Operators:

☑ These compare two operands and return a boolean result (true/1 or false/0).

` == ` (Equal to): Checks if two operands are equal.

` != ` (Not equal to): Checks if two operands are not equal.

` &gt; ` (Greater than): Checks if the first operand is greater than the second.

` &lt; ` (Less than): Checks if the first operand is less than the second.

` &gt;= ` (Greater than or equal to): Checks if the first operand is greater than or equal to the second.

` &lt;= ` (Less than or equal to): Checks if the first operand is less than or equal to the second.

```c
# include

int main() {

int x = 5, y = 10;
```

printf("x == y is %d\n", x == y);

printf("x != y is %d\n", x != y);

printf("x > y is %d\n", x > y);

printf("x < y is %d\n", x < y);

printf("x >= 5 is %d\n", x >= 5);

printf("y <= 5 is %d\n", y <= 5);

return 0;

}

```

☑ **3. Logical Operators:**

These combine or modify boolean expressions.

`&amp;amp;&amp;amp;` (Logical AND): Returns true if both operands are true.

`||` (Logical OR): Returns true if at least one operand is true.

`!` (Logical NOT): Reverses the logical state of its operand.

```c

# include

int main() {

int p = 1, q = 0; // 1 for true, 0 for false

printf("p && q is %d\n", p && q); // Output: 0

printf("p || q is %d\n", p || q); // Output: 1

```
printf("!p is %d\n", !p);   // Output: 0

return 0;

}

```

**4. Assignment Operators:**

☑ These assign a value to a variable.

  `=` (Simple Assignment): Assigns the value of the right operand to the left operand.

  `+=` (Add and Assign): `a += b` is equivalent to `a = a + b`.

  `-=` (Subtract and Assign): `a -= b` is equivalent to `a = a - b`.

  `=` (Multiply and Assign): `a = b` is equivalent to `a = a  b`.

  `/=` (Divide and Assign): `a /= b` is equivalent to `a = a / b`.

  `%=` (Modulo and Assign): `a %= b` is equivalent to `a = a % b`.

```c

```

include

```
int main() {

int val = 10;

printf("val = %d\n", val); // Output: 10

val += 5;

printf("val += 5: %d\n", val);

val -= 3;
```

```
printf("val -= 3: %d\n", val);

val = 2;

printf("val = 2: %d\n", val);

val /= 4;

printf("val /= 4: %d\n", val);

val %= 5;

printf("val %%= 5: %d\n", val);

return 0;

}
```

5. Increment and Decrement Operators:

☑ These increase or decrease the value of a variable by 1.

`++` (Increment): Increases the operand's value by 1.

Prefix increment (`++var`): Increments the value then uses it.

Postfix increment (`var++`): Uses the value then increments it.

`--` (Decrement): Decreases the operand's value by 1.

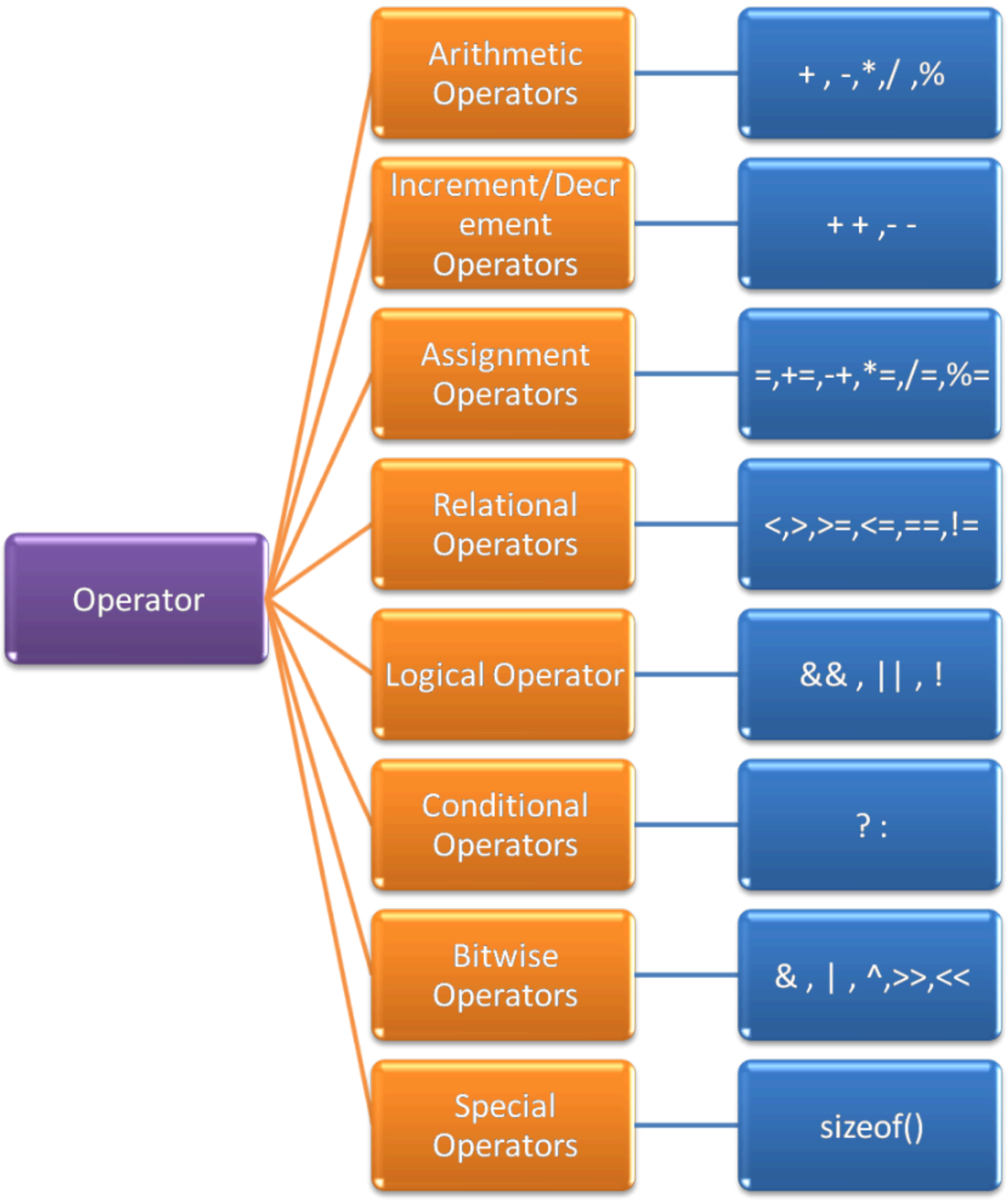
Prefix decrement (`--var`): Decrements the value then uses it.

Postfix decrement (`var--`): Uses the value then decrements it.

```c
```

include

```
int main() {
int i = 5, j = 5;
int result1, result2;
result1 = ++i; // i becomes 6, then result1 gets 6
printf("Prefix increment: i = %d, result1 = %d\n", i, result1);
//Output: i = 6, result1 = 6
result2 = j++; // result2 gets 5, then j becomes 6
printf("Postfix increment: j = %d, result2 = %d\n", j, result2);
//Output: j = 6, result2 = 5
return 0;
}
```


Operator
Arithmetic Operators
+ , -,*,/ ,%
Increment/Decrement Operators
+ + ,- -
Assignment Operators
=,+=,-+,*=,/=,%=
Relational Operators
<,>,>=,<=,==,!=
Logical Operator
&& , || , !
Conditional Operators
? :
Bitwise Operators
& , | , ^,>>,<<
Special Operators
sizeof()