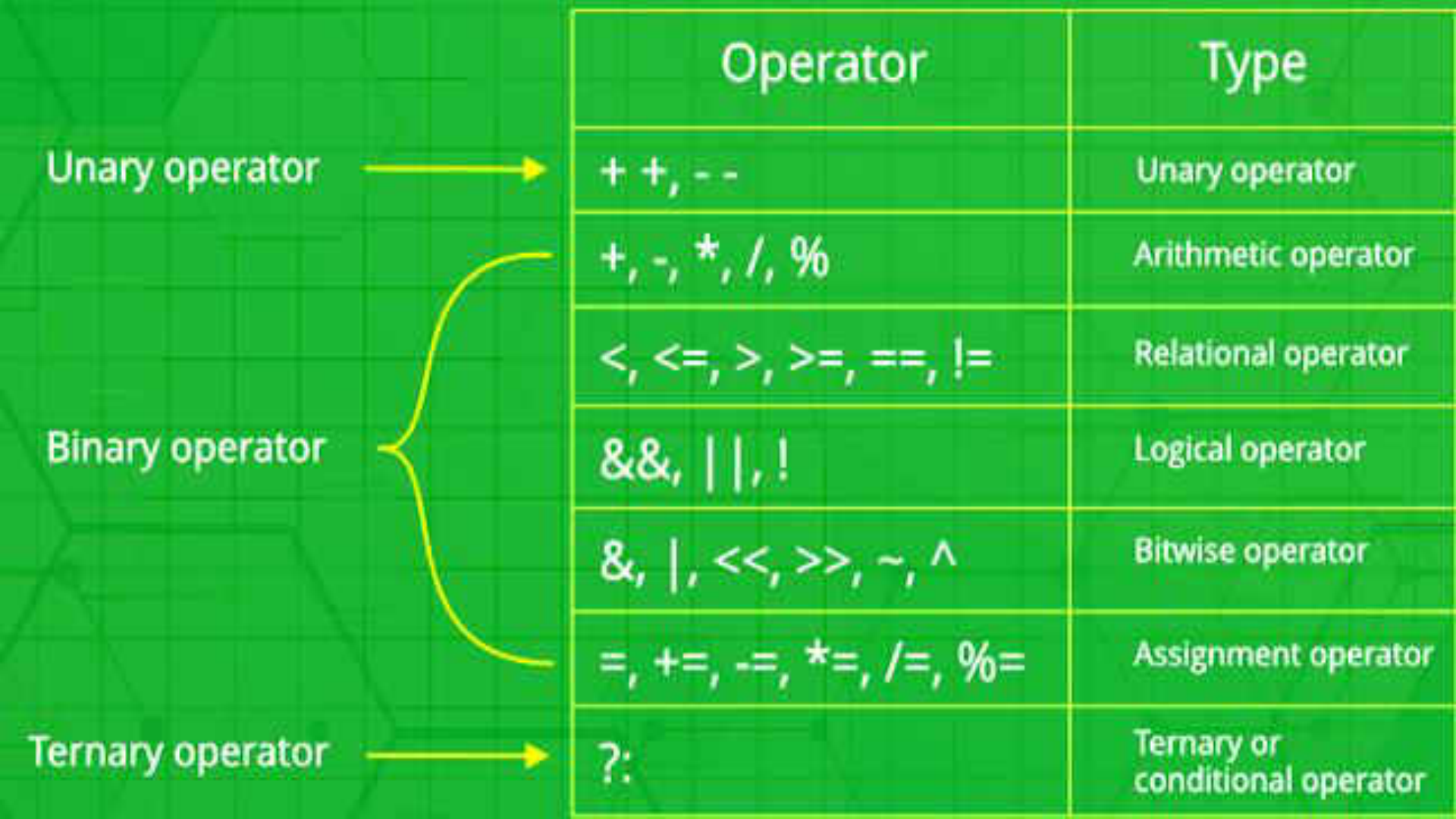
# 002 - C Programming Operators

# OPERATORS

## Operators:

- operators are used with operands to build expressions (or test and change values)


Operator
Type
Unary operator
+ +, - -
Unary operator
+, -, *, /, %
Arithmetic operator
Binary operator
<, <=, >, >=, ==, !=
Relational operator
&&, ||, !
Logical operator
&, |, <<, >>, ~, ^
Bitwise operator
=, +=, -=, *=, /=, %=
Assignment operator
Ternary operator
?:
Ternary or conditional operator

# increment and decrement operators

```
#include <stdio.h>
main()
{
int a = 10, b = 100;
printf("++a = %d \n", ++a);
printf("a++ = %d \n", a++);
printf("--b = %d \n", --b);
printf("b-- = %d \n", b--);
}
```

# C PROGRAM- Addition

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf(“Enter Your Marks: \n");
scanf("%d",&a);
b = a++;
printf(“Your New Marks are = %d",b);
getch();
}
```

# increment and decrement operators

- Increment (++) and Decrement (- -)
- Pre Increment is ++a
- (The value is incremented first and then transferred)
- Post Increment is a++
- (First Value is transferred and then value is incremented)
- Pre Decrement - -a
- (The value is decremented first and then transferred)
- Post Decrement a- -
- (First value is transferred and then value is decremented)

# C PROGRAM- Addition

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf(“Enter Your Marks: \n");
scanf("%d",&a);
b = a++;
printf(“Your New Marks are = %d",b);
getch();
}
```

# OPERATORS

- **Arithmetic operators (+ - / * % )**
- **Assignment operators (= * / % + - << >> & ^ I )**
- **Logical/Relational operators (== ! => < >= <= && || )**
- **Bitwise operators ( & | ^ << >> ~ )**

# Operators Priority Table

**Expression:** d=(a+b)/c

| Priority | Operator | Description |
|---|---|---|
| 1st Priority | *, /, % | Multiplication, Division, and Modulus division |
| 2nd priority | +, - | Addition and subtraction |
| 3rd Priority | = | Equal |

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf(“Enter Two No:\n");
scanf("%d %d",&a,&b);
c = a % b;
printf(“Answer = %d",c);
getch();
}
```

# Example : Arithmetic Operators

```
void main()
{
int a = 9,b = 4, c;
c = a+b;
printf("a+b = %d \n",c);
c = a-b;
printf("a-b = %d \n",c);
c = a*b;
printf("a*b = %d \n",c);
c = a/b;
printf("a/b = %d \n",c);
c = a%b;
printf("Remainder when a
        divided by b = %d \n",c);
}
```

# Program To Find Simple Interest

```
#include <stdio.h>
#include<conio.h)
void main()
{
  float p, r, si;
  int t;
clrscr();
  printf("Enter the values of principal, rate and time\n");
  scanf ("%f %f %d", &p, &r, &t);
  si = (p * r * t)/ 100;
  printf ("Amount = Rs. %f\n", p);
  printf ("Rate   = Rs. %f\n", r);
  printf ("Time   = %d years\n", t);
  printf ("Simple interest  = %f\n", si);   getch(); }
```

# Compound Assignment operators

| Operators | Example |
|---|---|
| =(Assignment) | C=A+B |
| +=(Add and assign) | C+=A(C=C+A) |
| -=(Subtract and assign) | C-=A(C=C-A) |
| *=(Multiply and assign) | C*=A (C=C*A) |
| /=(Divide and assign) | C/=A(C=C/A) |
| %=(Modulo and assign) | C%=A(C=C%A) |

# C Assignment Operators

```
#include <stdio.h>
#include<conio.h)
void main()  {
int a = 5, c;
clrscr();
c = a;
printf("c = %d\n", c);
c += a;
printf("c = %d\n", c);
c -= a;
printf("c = %d\n", c);
c *= a;
printf("c = %d\n", c);
c /= a;
printf("c = %d\n", c);
c %= a;
printf("c = %d\n", c);
getch();
}
```

# C Relational Operators

| Operator | Meaning of Operator | Example |
|---|---|---|
| == | Equal to | `5 == 3` is evaluated to 0 |
| > | Greater than | `5 > 3` is evaluated to 1 |
| < | Less than | `5 < 3` is evaluated to 0 |
| != | Not equal to | `5 != 3` is evaluated to 1 |
| >= | Greater than or equal to | `5 >= 3` is evaluated to 1 |
| <= | Less than or equal to | `5 <= 3` is evaluated to 0 |

# Relational Operators

```
void main()
{
int a = 5, b = 5, c = 10;
printf("%d == %d is %d \n", a, b, a == b);
printf("%d == %d is %d \n", a, c, a == c);
printf("%d > %d is %d \n", a, b, a > b);
printf("%d > %d is %d \n", a, c, a < c);
printf("%d < %d is %d \n", a, b, a < b);
printf("%d < %d is %d \n", a, c, a > c);
}
```

# Logical operator operation

| Operand (A) | Operand (B) | AND (A&B) | OR (A/B) | XOR (A^B) | NOT (~A) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

# Example : Relational Operators

```
main()
{
int a = 5, b = 5, c = 10, result;
result = (a == b) && (c > b);
printf("(a == b) && (c > b) is %d \n", result);
result = (a == b) && (c < b);
printf("(a == b) && (c < b) is %d \n", result);
result = (a == b) || (c < b);
printf("(a == b) || (c < b) is %d \n", result);
}
```

# Bitwise Operators

```
void main()
{
unsigned char a = 5, b = 9; // a = 5(00000101), b = 9(00001001)
     clrscr();
    printf("a = %d, b = %d\n", a, b);
    printf("a&b = %d\n", a & b);      // The result is 00000001
    printf("a|b = %d\n", a | b);       // The result is 00001101
    printf("a^b = %d\n", a ^ b);       // The result is 00001100
    printf("b<<1 = %d\n", b << 1);     // The result is 00010010
    printf("b>>1 = %d\n", b >> 1);     // The result is 00000100
    getch();
}
```

# Ternary operator

## Conditional or Ternary Operator (?:) in C/C++

**Syntax**

condition ? value_if_true : value_if_false

# Program to Find Largest Among Two Numbers Using Ternary Operator

```
#include <stdio.h>
void main()
{
int a = 5, b = 10, c;  // variable declaration
c = (a > b) ? a : b;  // Largest among n1 and n2
printf("Largest number between %d and %d is %d“,a, b, c);
}
```

# TYPE CONVERSION

```
void main()
{
int a=5, b=8, c;
float d, e;
c = a/b;
printf("\n %d",c);
d=a/b;
printf("\n %f \n",d);
e=float(a)/float(b);
printf("\n %f\n", e);
}
```

# The sizeof() operator

# sizeof( )

sizeof() call to display the size (in bytes) of the standard C data types (char,int, long…)

| | |
|---|---|
| sizeof (char) | =1 |
| sizeof (int) | =2 |
| sizeof (float) | =4 |
| sizeof (double) | =8 |

# sizeof() Operator

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i=20;
clrscr();
printf("value of i=%d\n",i);
printf("size of i=%d", sizeof(i));
getch();
}
```


thank you