Al-Mustaqbal University
College of Engineering & Technology
Biomedical Engineering Department

Computer

Lecture 4

Operators

PhD. BEng. Ahmed Hasan Janabi
PhD in Cybersecurity
Email: Ahmed.Janabi@uomus.edu.iq

Operators in C++

- The table below lists all possible operators that can be executed in C++ programming.

#	Types of C operator	Symbols of operators
1	Arithmetic operators	+ → Addition - → Subtraction * → Multiplication / → Division % → Modulo (remainder after division)
2	Logical operators	&& → Logical AND || → Logical OR ! → Logical NOT
3	Assignment operators	= → Assigns value of the left side to the right side += → a+=b is same as a = a+ b -= → a-=b is same as a = a- b *= → a*=b is same as a = a* b /= → a/=b is same as a = a/ b %= → a%=b is same as a = a% b

Operators in C++ (Cont…)

#	Types of C operator	Symbols of operators
4	Relational operators	< → Less than > → Greater than <= → Less than or equal to >= → Greater than or equal to == → is equal to != → is not equal to
5	Increment and Decrement operators	++ → increment value by 1 -- → decrement value by 1

6	Conditional operators	? → Example: x = (a>b)?a:b;
		it means
		If (a>b)
		x = a;
		else
		x = b;

7	Bitwise operators	& → bitwise AND | → bitwise OR ^ → bitwise exclusive-OR << → Shift Left (it multiples number by 2) >> → Shift Right (it divides number by 2)

Operators in C++ programming – Explanation

- Arithmetic operators are two types
  - Unary operators: operators that operate on a single operator

    Example

    b = a++;
  - Binary operators: operators that operate with two operators

    Example

    c = a+b;
    - '+' is the operator known as addition operator,
    - 'a' and 'b' are operands
    - The addition operator tells the compiler to add both of the operands 'a' and 'b' and put them in c variable

Logical operators:

- The result of the operation of a logical operator is a **Boolean** value either *true* or *false*.
- used to test one or more condition or make decisions

Operators in C++ programming – Explanation

- Relational operators are used to compare two numbers and taking decisions based on their relation.
- Relational expressions are used in decision statements such as ***if*** , ***for***.
- Conditional operator: It takes three arguments, **condition? exp1:exp2** if condition is true then execute exp1 otherwise exp2 will be executed.
- **Bitwise Operators**
  - Bitwise operators are used to perform operation bit by bit.
  - Bitwise operators may not be applied to float or double.

Assignment Operators in C++

- The Assignment operators in C++ are some of the C Programming Operator, which are useful to assign the values or the result of an expression to the declared variables.
- The equals (=) operator is the most commonly used assignment operator in C++.
- For example:

```
int i = 10;
```

Arithmetic operators – Example

```c
#include <stdio.h>

int main() {

   int a = 9, b = 4, c;

   c = a+b;                          //addition + → (a+b=13)
   printf("a+b = %d \n", c);

   c = a-b;                          //subtraction - → (a-b=5)
   printf("a-b = %d \n", c);

   c = a*b;                          //multiplication * → (a*b=36)
   printf("a*b = %d \n", c);

   c = a/b;                          //division / → (a/b=2)
   printf("a/b = %d \n", c);

   c = a%b;                          //modulo % → (a%b=1)
   printf("Remainder when a divided by b = %d \n", c);

return 0;
}
```

Assignment Operators in C

- The below table displays all the assignment operators present in C Programming with an example.

C Assignment Operators	Example	Explanation
=	x = 25	Value 25 is assigned to x
+=	x += 25	This is same as → x = x + 25
–=	x –= 25	This is same as → x = x – 25
*=	x *= 25	This is same as → x = x * 25
/=	x /= 25	This is same as → x = x / 25
%=	x %= 25	This is same as → x = x % 25

Assignment Operators in C++ – Example

- In following C++ assignment operators Program, two integer variables a and Total are used. And their values are 7 and 21, respectively. It shows the working functionality of all the Assignment Operators in C++ Programming Language.

```
/* Program for Assignment Operators in C*/
#include <stdio.h>

int main()
{
  int a = 7;
  int Total = 21;

   printf(" Value of the Total = %d \n", Total += a );
   printf(" Value of the Total = %d \n", Total -= a );
   printf(" Value of the Total = %d \n", Total *= a );
   printf(" Value of the Total = %d \n", Total /= a );
   printf(" Value of the Total = %d \n", Total %= a );

    return 0;
  }
```

Assignment Operators in C++ – Example

```c
/* Program for Assignment Operators in C*/
#include <stdio.h>

int main()
{
   int a = 7;
   int Total = 21;

    printf(" Value of the Total = %d \n", Total += a );
    printf(" Value of the Total = %d \n", Total -= a );
    printf(" Value of the Total = %d \n", Total *= a );
    printf(" Value of the Total = %d \n", Total /= a );
    printf(" Value of the Total = %d \n", Total %= a );

     return 0;
   }
```

output

```
Value of the Total = 28
Value of the Total = 21
Value of the Total = 147
Value of the Total = 21
Value of the Total = 0
```

Note:

- `\n` : means new line
- `\t` : means take a tab space then write

Assignment Operators in C++ – Example

- The ***printf*** statements will perform C++ Programming Assignment operations on **a** and **Total** and then display the output (result).
- Let us see the C Programming Operator functionality in this C Program

```
printf(" Value of the Total = %d \n ", Total += a );
```

Total += a

means → Total = Total + a = 21 + 7 = 28

```
printf(" Value of the Total = %d \n ", Total -= a );
```

Total -= a

means → Total = Total - a = 28 - 7 = 21

```
printf(" Value of the Total = %d \n ", Total *= a );
```

Total *= a

means → Total = Total * a = 21 * 7 = 147

Assignment Operators in C – Example

```
printf(" Value of the Total = %d \n ", Total /= a );
```

Total /= a

means →Total = Total / a = 147 / 7 = 21

```
printf(" Value of the Total = %d \n ", Total %= a );
```

Total %= a

means →Total = Total % a = 21 % 7 = 0 (the remainder of 21/7 is 0)

Increment and Decrement – Explanation

❖ Increment → ++

- **a++** is postfix, the expression is evaluated first and then the value is incremented.

  Example

  a = 10;

  b = a++; // after this statement, a = 11, b = 10

- **++a** is prefix, the value is incremented first and then the expression is evaluated.

  Example

  a = 10;

  b =++a; // after this statement, a = 11, b = 11

Increment and Decrement – Explanation

- **Decrement → –**
  - **a--** is postfix, the expression is evaluated first and then the value is decremented.

    Example

    a = 10;

    b = a--; // after this statement, a = 9, b = 10

  - **--a** is prefix, the value is decremented first and then the expression is evaluated.

    Example

    a = 10;

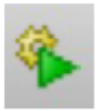
    b = --a; // after this statement, a = 9, b = 9

Class #2 – Adding two double numbers

```c
#include <stdio.h>
 int main()
{
    double a, b, Sum = 0;

    printf("Enter the 1st number: ");
    scanf("%lf", &a);


    printf("Enter the 2nd number: ");
    scanf("%lf", &b);


    Sum = a + b;
    printf("%lf + %lf = %lf", a, b, Sum);

     return 0;
}
```

Output:

?



Build and run

```
Faris.Alghareb@Users-MacBook-Pro ~ % '/Users/user/Documents/
Enter the 1st number: 5.5
Enter the 2nd number: 3.2
5.500000 + 3.200000 = 8.700000
Faris.Alghareb@Users-MacBook-Pro ~ %
```

Area calculation of a circle

Write a C program that reads the radius of a circle and and prints its area and circumference.

Ans:

```c
#include <stdio.h>

int main() {

?

return 0;
}
```

THANK YOU