Covered Topics Under UNIT-1 of "PPS-PROGRAMMING FOR PROBLEM SOLVING (BCS101 / BCS201)"

shwetatiwari08@recabn.ac.in
shwetatiwari08aug@gmail.com

Class Notes

Published Date: November, 2022

PPS: UNIT-1

Introduction to Components of a Computer System

FALL SEMESTER, YEAR (I/II sem, 1st yr)

FALL SESSION (2022-23)
(PPS)
MS. SHWETA TIWARI
Published Date: November, 2022

shwetatiwari08@recabn.ac.in
shwetatiwari08aug@gmail.com

TOPIC On : UNIT-1: Operators and Expression in C Programming

—

By SHWETA TIWARI

Under On: Introduction to Components of a Computer System

PREPARED FOR
Engineering Students
All Engineering College

PREPARED BY
SHWETA TIWARI

TOPIC On : ***UNIT-1:*** Operators and Expression in C Programming

Operators and Expression in C Programming

There are 6 operators in C Programming : Arithmetic Operator, Relational Operator, Logical Operator Increment, Decrement Operators, Assignment Operators, Conditional or Ternary Operators and Bitwise Operators.

1. Arithmetic operators
2. Relational operator
3. Logical operator
4. Increment and Decrement operator
5. Assignment operator
6. Conditional or Ternary Operators
7. Bitwise operator

1. *Arithmetic operators*

Arithmetic Operators are used to perform basic arithmetic operations like addition, subtraction, division etc.

Operators	Meaning	Example	Result
+	Addition	4 + 2	6
–	Subtraction	4 – 2	2
*	Multiplication	4 * 2	8
/	Division	4 / 2	2
%	Modulus operator to get remainder in integer division	5 % 2	1

Library functions (Intrinsic Function or Math Functions)

Mathematical Notation	C function	Meaning
$\sqrt{a}$	sqrt (a)	Square root of a
$\|a\|$	fabs (a)	Absolute value of a
e^x	exp (x)	Exponential of x
$\log x$	log (x)	Logarithm of x
x^y	pow (x, y)	x raised to y
$\sin x$	sin (x)	Sine of x (x in radians)
$\cos x$	cos (x)	Cosine of x (x in radians)
$\tan x$	tan (x)	Tangent of x (x in radians)
	ceil (x)	Round off a float value to the nearest integer upward (e.g. 5.34 will be rounded off to 6.0)
	floor (x)	Rounded off a float value to the nearest integer downwards (e.g. 5.34 will be rounded off to 5.0)

2. ***Relational operator***

Relational operators are used to compare the value of operands (expressions) to produce a logical value. A logical value is either True or False.

Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True

3. ***Logical operator***

Logical operators are used to connect more relational operations to form a complex expression called logical expression. A value obtained by evaluating a logical expression is always logical, i.e. either True or False.

Operator	Meaning	Example	Result
&&	Logical and	(5<2) and (5>3)	False
||	Logical or	(5<2) or (5>3)	True
!	Logical not	not (5<2)	True

5. Increment or Decrement Operators, Assignment Operators, Conditional or Ternary Operators and Bitwise Operators

4. ***Increment and Decrement operator***

Operators	Meaning
++a	Increment a by 1, then use the new value of a
a++	Use value of a, then increment a by 1
--b	Decrement b by 1, then use the new value of b
b--	Use the current value of b, then decrement by 1

Ex : -

```
a = 15;                    b = 15;
a++ or ++a                 b-- or --b
result = 16                Result = 14
```

5. ***Assignment operator***

Assignment operators are used to perform arithmetic operations while assigning a value to a variable.

Operator	Example	Equivalent Expression (m=15)	Result
=	y = a+b	y = 10 + 20	30
+=	m +=10	m = m+10	25
-=	m -=10	m = m-10	5
*=	m *=10	m = m*10	150
/=	m /=10	m = m/10	1.5
%=	m %=10	m = m%10	5

6. ***Conditional or Ternary Operators***

   Variable=(Condition)? Value 1: Value2;

   Eg: big=(a>b)?a:b;

   if(a>b)

   big=a;

   else

   big=b;

7. ***Bitwise operator***

   Bitwise operators are used to perform operations at binary digit level. These operators are not commonly used and are used only in special applications where optimized use of storage is required.

Operator	Meaning
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR / Bitwise XOR
~	Bitwise inversion (one's complement)
<<	Shifts the bits to left / Bitwise Left Shift
>>	Shifts the bits to right / Bitwise Right Shift