



CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

# BTES104 Computer Programming in C

Avinash Y. Surywanshi

KCES's College of Engineering and Management,  
Jalgaon - 425001, Maharashtra, India

February 26, 2021



# BTES104 Computer Programming in C

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

## 1 Unit II :

## 2 Constants, Variables and Keywords

- C Constants
- C Variables
- C Keywords

## 3 Data Types

## 4 Operators

- Arithmetic operators
- Relational operators
- Logical operators
- Increment and Decrement operators
- Assignment operators
- Bitwise Operator
- Conditional operators

## 5 Precedence and Order of Evaluation



CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

## 1 Unit II :

### 2 Constants, Variables and Keywords

- C Constants
- C Variables
- C Keywords

### 3 Data Types

### 4 Operators

- Arithmetic operators
- Relational operators
- Logical operators
- Increment and Decrement operators
- Assignment operators
- Bitwise Operator
- Conditional operators

### 5 Precedence and Order of Evaluation



CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

## 1 Unit II :

## 2 Constants, Variables and Keywords

- C Constants
- C Variables
- C Keywords

## 3 Data Types

## 4 Operators

- Arithmetic operators
- Relational operators
- Logical operators
- Increment and Decrement operators
- Assignment operators
- Bitwise Operator
- Conditional operators

## 5 Precedence and Order of Evaluation



# Constants, Variables and Keywords

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- The alphabets, numbers and special symbols when properly combined form constants, variables and keywords.
- A constants is an entict that doesn't change whereas a variable is an entity that may change.



# Types of C Constants

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

C constants can be divided into two major categories:

- 1 Primary Constants
- 2 Secondary Constants



# C Constants

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

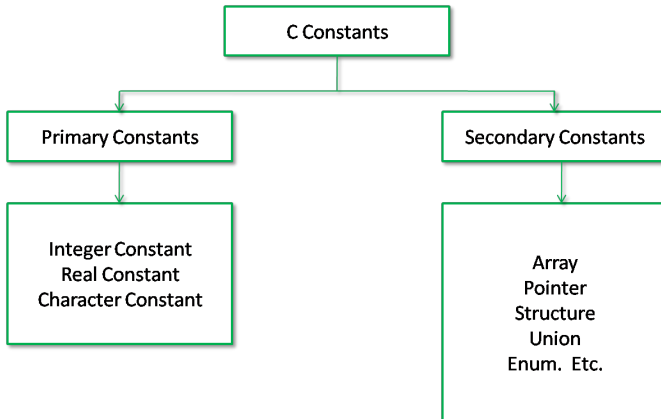
Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of





# Rules for Integer Constants

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- 1 An integer constant must have at least one digit.
- 2 It must not have a decimal point.
- 3 It can be either positive or negative.
- 4 If no sign precedes an integer constant it is assumed to be positive.
- 5 No commas or blanks are allowed within an integer constant.
- 6 The allowable range for integer constants is -32768 to 32767(for 16 bit compiler like Turbo C or Turbo C++)

Example: 426

+278

-8000





# Rules for Real Constants

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- 1 A real constant must have at least one digit.
- 2 It must have a decimal point.
- 3 It Could be either positive or negative.
- 4 Default Sign is Positive.
- 5 No commas or blanks are allowed within an real constant.

Example: 426.0

-32.78

-48.698



# Rules for real Constants in Exponential form

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- 1 The Mantissa part and the exponential part should be separated by a letter e.
- 2 The mantissa part may have a positive or negative sign.
- 3 Default sign of mantissa part is positive.
- 4 The exponent must have a at least one digit, which must be a positive or negative integer. Default sign is positive.
- 5 Range of real constants expressed in exponential form is  $-3.4e38$  to  $3.4e38$ .

Example:  $+3.2e-5$

$3.2e7$

$-5.2e-8$



# Rules for Character Constants

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

- 1 The maximum length of a character constant can be one character.
- 2 The Character should be included in a single inverted comma both pointing to the left.

Example: 'A'



# What is Variables

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

**C Variables**

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- An entity that may vary during program execution is called a variable.
- Variables names are names given to location in memory.
- Memory location can contain integer, real or character constants.
- A particular type of variable can hold only the same type of constant



# Rules for constructing variable names

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

**C Variables**

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- 1 A variable name is any combination of 1 to 31 alphabets , digits or underscores.
- 2 The first character in the variable name should be an alphabet.
- 3 No commas or blank are allowed within a variable name.
- 4 No special symbol other than an underscore can be used in a variable name.
- 5 The variable name should not be a keyword.
- 6 The Variable name is case sensitive.

Example: si\_int  
m\_phr  
stu\_e\_16



# C Keywords

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

All the words that have a predefined meaning in c are called as keywords. There are 32 keywords available in C

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

1

Unit II :

2

Constants, Variables and Keywords

- C Constants
- C Variables
- C Keywords

3

Data Types

4

Operators

- Arithmetic operators
- Relational operators
- Logical operators
- Increment and Decrement operators
- Assignment operators
- Bitwise Operator
- Conditional operators

5

Precedence and Order of Evaluation



# Data Types

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

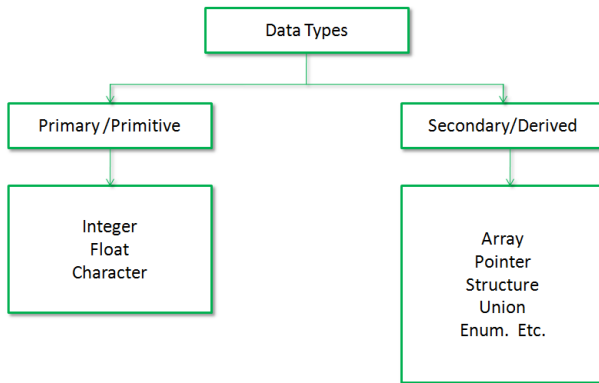
Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

All the words that have a predefined meaning in c are called as keywords. There are 32 keywords available in C







# Rules for Integer Data Type

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

- An integer constant must have at least one digit.
- It must not have a decimal point.
- It can be either positive or negative.
- If no sign precedes an integer constant it is assumed to be positive.
- No commas or blanks are allowed within an integer constant.
- The allowable range for integer constants is -32768 to 32767 (for 16 bit compiler like Turbo C or Turbo C++)
- Consumes 2 bytes of memory space for variable declare
- Format specifier is %d.



# Types of Integer Data Type

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- Integer can be of another type called as short and long, both are keywords.
- short int (2 bytes required, format specifier is %d) and
- long int (4 bytes required format specifier is %ld, -2147483648 to +2147483647)

Example: `int a=10, bigger;`  
`short int big;`  
`long int biggest;`



# Rules for Float

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

- A real constant must have at least one digit.
- It must have a decimal point.
- It Could be either positive or negative.
- Default Sign is Positive.
- No commas or blanks are allowed .
- Consumes 4 bytes of memory space
- Format specifier is %f
- The keyword used to define floating point values is float



# Types of FLoat Data Type

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- Float can be of another type called as double and long double, both are keywords.
- double (8 bytes required, format specifier is %lf, -1.7e308 to +1.7e308) and
- long double (10 bytes required format specifier is %Lf, -1.7e4932 to +1.7e4932).

Example: float c=11.50,  
small; double medium;  
long double larger;



# Rules for Character data type

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- Consumes 1 byte of memory space.
- Format specifier is %c
- The maximum length of a character constant can be one character.
- The Character should be included in a single inverted comma both pointing to the left.

Example: 'A'



# Secondary/Derived data type

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

**Array:** It is collection of objects of same type, Array elements stored in continuous memory Location.

Ex. `int a [5]; /*array of 5 elements whose name is 'a' and type is integer */`

**Structure:** It is collection of objects of different type.

Ex. `struct student  
{ int rollno=10;  
float marks=10.5;  
};`



# Secondary/Derived data type

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

**Union:** It is collection of objects of different type. The difference between structure and union is that, union uses shared memory for variables while structure assigns separate memory to variables

**Enumeration:** It is user defined data type allow user to create own data type values.

Ex. enum day {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday};



# Defining Symbolic Constants

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- Names given to values that cannot be changed.
- Implemented with the `# define` preprocessor directive.
- Examples: `#define N 3000`, `#define FALSE 0` `#define PI 3.14159`
- Before the program compiled, the `#` preprocessor directive will replace the symbol names with their values which is defined previously.





# Defining Symbolic Constants

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

- All # statements are processed first, and the symbols (like N) which occur in the C program are replaced by their value (like 3000). Once this substitution has taken place by the preprocessor, the program is then compiled.
- Note that preprocessor statements begin with a # symbol, and are not terminated by a semicolon.



CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

## 1 Unit II :

## 2 Constants, Variables and Keywords

- C Constants
- C Variables
- C Keywords

## 3 Data Types

## 4 Operators

- Arithmetic operators
- Relational operators
- Logical operators
- Increment and Decrement operators
- Assignment operators
- Bitwise Operator
- Conditional operators

## 5 Precedence and Order of Evaluation



# Operators used in C programming language

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

**Operators**

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

The following are the different types of the operators

- Arithmetic operators
- Relational operators
- Logical operators
- Increment and Decrement operators
- Assignment operators
- Bitwise Operator
- Conditional operators



# Arithmetic operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division (Gives <b>Quotient</b> after division)
%	Modulus (Gives <b>Remainder</b> after division)



# Arithmetic operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

**Arithmetic operators**

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Ex. float a=4.0,b=2.0, c;  
c=a/b; /\* gives output 2.0 because quotient is 2.0\*/  
c=a%b; /\* gives output 0 because remainder is zero \*/



# Relational operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

**Relational operators**

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Operator	Meaning
<	Less than
>	Greater than
<=	Less than equal to
>=	Greater than equal to
==	Equal to
!=	Not equal to

**Note:** Here confusion may occur in equal to operator (==) and assignment operator (=) both having different meaning in c programming



# Logical operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

**Logical operators**

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Operator	Meaning
<b>&amp;&amp;</b>	Logical AND
<b>  </b>	Logical OR
<b>!</b>	Logical NOT



# Logical operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

**Logical operators**

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Operator	Symbol	Usage	Operation
LOGICAL AND	&&	<b>exp1</b> && <b>exp2</b>	Requires both <b>exp1</b> and <b>exp2</b> to be TRUE to return TRUE. Otherwise, the logical expression is FALSE.
LOGICAL OR		<b>exp1</b>    <b>exp2</b>	Will be TRUE if either (or both) <b>exp1</b> or <b>exp2</b> is TRUE. Otherwise, it is FALSE.
LOGICAL NOT	!	! <b>exp</b>	Negates (changes from TRUE to FALSE and visa versa) the expression.





# Increment and Decrement operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

**Increment and  
Decrement operators**

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Operator	Meaning
<b>++</b>	Increment
<b>--</b>	Decrement



# Increment and Decrement operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
**Increment and  
Decrement operators**  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

In C, specialized operators have been set aside for the incrementing and decrementing of **integer variables**.

The increment and decrement operators are **++** and **--** respectively they will increment or decrement the value of associated variable by **1** only.

The increment / decrement operators may be of prefix (before variable) or postfix form (after variable)

**Prefix form:**

**++i;**

**--i;**

**Postfix form:**

**i++;**

**i--;**

Ex. Assuming i as variable

is equivalent to

is equivalent to

is equivalent to

is equivalent to

**i=i+1;**

**i=i-1;**

**i=i+1;**

**i=i-1;**



# Assignment operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

**Assignment  
operators**

Bitwise Operator

Conditional operators

Precedence  
and Order of

Operator	Meaning
<b>=</b>	Value Assignment
<b>a+=b</b>	$a = a + b$
<b>a-=b</b>	$a = a - b$
<b>a*=b</b>	$a = a * b$
<b>a/=b</b>	$a = a / b$
<b>a%=b</b>	$a = a \% b$



# Assignment operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

**Assignment  
operators**

Bitwise Operator

Conditional operators

Precedence  
and Order of

## General Syntax

**Variable name = expression;**



# Bitwise Operator

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

**Bitwise Operator**

Conditional operators

Precedence  
and Order of

Operator	Meaning
<b>&amp;</b>	Bitwise <b>AND</b>
<b> </b>	Bitwise <b>OR</b>
<b>^</b>	Bitwise <b>XOR</b>
<b>~</b>	Bitwise <b>Complement</b>
<b>&lt;&lt;</b>	Left Shift
<b>&gt;&gt;</b>	Right Shift



# Bitwise Operator

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Input		Output		
a	b	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Figure: Truth table for AND, OR, XOR



# Bitwise Operator

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Ex.            `int a=12, b=8,c;`  
                 `c=a&b;`

This will perform AND operation like as follows:

<code>a = 12</code>	<code>=</code>	<code>00001100</code>
<code>b = 8</code>	<code>=</code>	<u><code>&amp;00001000</code></u>
<code>c = 8</code>	<code>=</code>	<code>00001000</code>

Both operands must be of same data-type

**AND** operator used to **check** particular bit is **ON/OFF**

**OR** operator is used to **put** particular bit **ON**

Figure: Truth table for AND, OR, XOR



# Right Shift Operator

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

**Right Shift (>>)** - It will shift the bits of operand towards **right**;  
the blanks are created on left side which are filled by zero(s).

In general declared as

**Variable >> number of bits**

**Ex.** `int a;` /\*assume we have given value a=8 \*/

`a>>1;`

					1024	512	256	128	64	32	16	8	4	2	1
$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

$8 = 00000000000001000$    $0000000000000100 = 4$   






# Right Shift Operator

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

**Bitwise Operator**

Conditional operators

Precedence  
and Order of

**Ex.** if the variable **a** contains the bit pattern 11010111,  
then, **a >> 1** would give 01101011  
and **a >> 2** would give 00110101.



# Left Shift Operator

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

**Left Shift (<<)** - It will shift the bits of operand towards **left**;  
the blanks are created on right side which are filled by zero(s).  
In general declared as:

**Variable << number of bits**

**Ex.** int a; /\*assume we have given value a=2 \*/

A<<1;

8 = 0000000000000100



0000000000000010 = 2



# Conditional operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Operator	Meaning
<b>? :</b>	<b>Conditional Operator</b>



# Conditional operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

General Syntax
<b>(Condition)? true : False</b>
<b>or</b>
<b>(Expression1)? Expression2 : Expression3</b>



# Conditional operators

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Ex.     `int a=10,b=5;`  
         `(a>b)? printf("a is larger") : printf("b is larger");`

In this case output will be  
**a is larger**



CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants  
C Variables  
C Keywords

Data Types

Operators

Arithmetic operators  
Relational operators  
Logical operators  
Increment and  
Decrement operators  
Assignment  
operators  
Bitwise Operator  
Conditional operators

Precedence  
and Order of

## 1 Unit II :

## 2 Constants, Variables and Keywords

- C Constants
- C Variables
- C Keywords

## 3 Data Types

## 4 Operators

- Arithmetic operators
- Relational operators
- Logical operators
- Increment and Decrement operators
- Assignment operators
- Bitwise Operator
- Conditional operators

## 5 Precedence and Order of Evaluation



# Precedence and Order of Evaluation

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Operator	Associatively
<b>!</b>	<b>Logical NOT</b>
<b>++ --</b>	<b>R to L</b>
<b>* / %</b>	<b>L to R</b>
<b>+ -</b>	<b>L to R</b>
<b>&lt; &gt; &lt;= &gt;=</b>	<b>L to R</b>
<b>== !=</b>	<b>L to R</b>
<b>&amp;&amp;</b>	<b>L to R</b>
<b>  </b>	<b>L to R</b>
<b>=</b>	<b>R to L</b>



# Evolution of expressions

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

## Arithmetic expressions

Ex. Example shows how the following expression is evaluated

**$1 + 2 * 3 - 4$**

**$1 + 6 - 4$**

**$7 - 4$**

**3**

Expressions enclosed in parentheses are evaluated first.

Example:

**$(1 + 2) * (3 - 4)$**

**$3 * -1$**

**-3**





# Evolution of expressions

CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

Suppose we want to write expression  $2x-3y$  then,

In C programming it can be written as

$$(2*x) - (3*y)$$

Here the brackets are evaluated first then

Further operation is performed



CP

A. Y.  
Surywanshi

Outline

Unit II :

Constants,  
Variables and  
Keywords

C Constants

C Variables

C Keywords

Data Types

Operators

Arithmetic operators

Relational operators

Logical operators

Increment and  
Decrement operators

Assignment  
operators

Bitwise Operator

Conditional operators

Precedence  
and Order of

*Thank You...*

*Email : [avin\\_surya@yahoo.com](mailto:avin_surya@yahoo.com)*