Unit 04 & 05: deepankar

Saturday, October 21, 2023 7:06 PM

operators and control structures

- → operators and expressions in C
- → I/O functions
- → formatted and unformatted
- → conditional & branching
- → if else, for, while, do-while
- → jump statements → break, continue
- → goto & return

expression → combination of operands & operators
- operands → operation is performed on them
- operators → symbols that perform some operation

Bitwise operators

a = 10, b = 5

a = 1010
b = 0101

Operator	Description	Example
&	Bitwise AND	a & b # 0 0000
|	Bitwise OR	a | b # 15 1111
^	Bitwise XOR	a ^ b # 15 1111
~	1's complement	~a # some negative value
<<	left shift	a << 1 # 20 10100
>>	right shift	a >> 1 # 5 101

assuming 8 bit number

10 → 0000 1010

1's → 1111 0101

(-117)

1. Arithmetic operators (+, -, *, /, %, ++, --)
2. Relational operators (==, !=, >, <, >=, <=)
3. Logical operators (&&, ||, !)
4. Bitwise operator (&, |, ^, ~, <<, >>)
5. Assignment operator (=, +=, -=, *=, /=, <<=, >>=, &=, ^=, |=)
6. Miscellaneous operators (sizeof(), condition ? X : Y (ternary operator), , (comma operator), . (dot) → (arrow) (member operators), & (reference operator), * (dereference operator))
   - → cast operator (typecasting) int (5.998); # 5

Left shift & assignment

C <<= 2

Precedence of operators

1	()	parenthesis	
	[]	square brackets	
	.	dot operator	
	→	arrow operator	
	++, --	post increment, decrement	
	++, --	pre increment, decrement	
	+, -	unary plus & minus	
	!, ~	Logical NOT, Bitwise complement	
2	*	dereferencing operator	
	&	referencing	
	sizeof	determining size in bytes	
3	*, /, %		
4	+, -		
5	>>, <<	Bitwise shift	
6	<<, <<=		
7	>>, >>=		
8	&	Bitwise &	
9	^	Bitwise ^	
10	|	Bitwise |	
11	&&		
12	||		
13	? :	ternary operator	
14	=, +=, -=, *=, /=, >>=, <<=	assignment	
15	,	comma	

Types of expressions

1. Arithmetic expressions
2. Relational expressions
3. Logical expressions
4. Conditional expressions

Input Output Streams

<stdio.h>
- → stdin # input stream
- → stdout # output stream
- → stderr # error stream

formatted and unformatted functions

Formatted
- Input: scanf()
- output: printf()

Unformatted
- Input: getch(), getche(), getchar(), gets()
- Output: putch(), putchar(), puts()

Control Structure

1. Sequence Structures (sequence)
2. Branching structures used for selection (selection) → if-else (2-way branch), switch (multiple branches)
3. Looping structures for repetation or iteration (loops) → for, while (entry control), do-while (exit control)
4. Jump statement → break, continue, goto, return

format specifiers

Integers
- short signed %d, %I
- short unsigned %u
- long signed %ld
- long unsigned %lu
- unsigned hexadecimal %x
- unsigned octal %o

Real
- float %f
- double %d

String
- string %s

Character
- signed character %c
- unsigned character %c

	Escape Sequence	Name
1	\n	new line
2	\r	carriage return
3	\t, \v	horizontal tab, vertical tab
4	\f	form feed
5	\a	alarm / beep
6	\b	backspace
7	\\	backlash character