Anil Neerukonda Institute of Technology & Sciences
Department of Computer Science & Engineering (AI & ML, DS)

CSE117 Problem Solving with C

Handout - Lab Session - 3

Objective:

- To understand the logical operators: and, or, and not.
- To understand how a C program evaluates a logical expression.
- To write C programs using logical and comparative operators.
- To write C programs that use two-way selection: if…else statements.

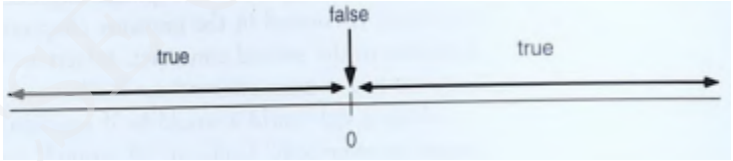
Pre-Lab: Go through the concepts of logical and comparative operators and two-way selection. Write the algorithm and flowcharts for all the exercise problems given in this handout.

During Lab: Solve all the exercise problems. You should work on the additional set of programs only after completing this week's tasks.

Post Lab: Take the quiz.

Logical Data

A piece of data is called logical if it conveys the idea of true or false. C had no logical data type; C programmers used other data types, such as int, to represent logical data. If the data value is zero, it is considered false. If it is nonzero, it is considered true. This concept of true and false on a numeric scale is seen below



Logical Operators

C has three logical operators for combining logical values and creating new logical values: ***not***, ***and***, and ***or.***

- ➢ not operator
  The not operator (!) is a unary operator with precedence 15. It changes a true value to a false and a false value to a true.
- ➢ and operator
  The and operator (&&) is a binary operator with a precedence of 5. Since it is a binary operator, four distinct combinations of values in its operands are possible. The result is true only when both operands are true, it is false in all other cases.

➢ or operator

The or operator (| |) is a binary operator with a precedence of 4. Since it is a binary operator, four distinct combinations of values in its operands are possible. The result is false only when both operands are false, it is true in all other cases.

not operator	
x	**!x**
true	false
false	true

and operator		
x	**y**	**x && y**
false	false	false
false	true	false
true	false	false
true	true	true

or operator		
x	**y**	**x || y**
false	false	false
false	true	true
true	false	true
true	true	true

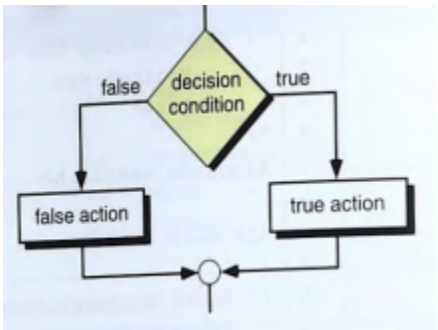
Comparative Operators

C language provides six comparative operators. Comparative operators are divided into two categories, relational and equality. They are all binary operators that accept two operands and compare them. The result is a boolean type - true or false.

Type	Operator	Meaning	Precedence
Relational	<	less than	10
	<=	less than or equal	
	>	greater than	
	>=	greater than or equal	
Equality	= =	equal	9
	!=	not equal	

Two-way Selection

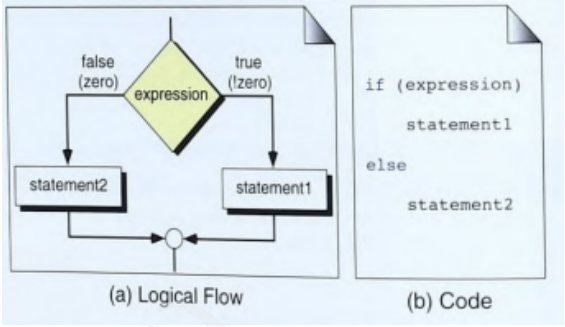
The basic decision statement in the computer is the two-way selection. The decision is described to the computer as a conditional statement that can be answered either true or false. If the answer is true, one or more action statements are executed. If the answer is false, a different action or set of actions is executed. Regardless of which set of actions is executed, the program



continues with the next statement after the selection. The flowchart for two-way decision logic is shown in the figure.

if … else

C implements two-way selection with the if. . .else statement. An if. . . else statement is a composite statement used to make a decision between two alternatives. The figure here shows the logic flow for an if … else statement. The expression can be any C expression. After it has been evaluated, if its value is true, statement1 is executed: otherwise, statement2 is executed. It is impossible for both statements to be executed in the same evaluation.

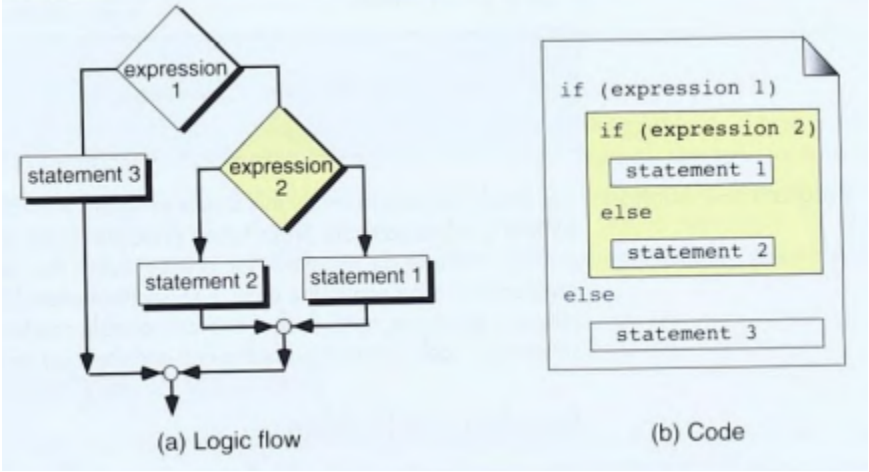


The syntactical points we must remember about if-else statements are summarized in the table given here.

1. The expression must be enclosed in parentheses.
2. No semicolon (;) is needed for an *if…else* statement; statement 1 and statement 2 may have a semicolon as required by their types.
3. The expression can have a side effect.
4. Both the true and the false statements can be any statement (even another *if…else* statement) or they can be a null statement.
5. Both statement 1 and statement 2 must be one and only one statement. Remember, however, that multiple statements can be combined into a compound statement through the use of braces.
6. We can swap the position of statement 1 and statement 2 if we use the complement of the original expression.

Nested if statements

When an if...else is included within an if ..else, it is known as a nested if statement. This figure shows a nested if statement. There is no limit to how many levels can be nested, but if there are more than three, they can become difficult to read.



** Refer to the textbook for more information. (Chapter 5, A structured programming approach using C. Behrouz A. Forouzan & Richard F. Gilberg)

Lab Exercises

Exercise 1: If x=3, y=0, z=-4, print the values of the following expressions.

i. x && y | | z

ii. x | | y && z

iii. (x && y) | | z

iv. (x | | y) && z

v. (x && z) | | y

Exercise 2: Read the score from the user at the keyboard. Write an if statement to assign the value 1 to the variable *pass* if the integer variable *score* is 40 or greater. Print the pass status of the student.

Sample Test Cases	Input	Output
Test Case 1	Enter your score: 50	Pass - 1 Congratulation, you passed !!
Test Case 2	Enter your score: 30	Pass - 0 Sorry, you did not pass !!

Exercise 3: Read one number from the user at the keyboard. Write if-else statements to print if the number is even or odd.

Sample Test Cases	Input	Output
Test Case 1	Enter the number: 20	The number, 20 is an even number.
Test Case 2	Enter the number : 17	The number, 17 is an odd number.

Exercise 4: Read an alphabet from the user at the keyboard. Write if-else statements to print if the alphabet is a vowel or consonant.

Sample Test Cases	Input	Output
Test Case 1	Enter the alphabet: a	The alphabet, a is a vowel.
Test Case 2	Enter the alphabet: b	The alphabet, b is a consonant.

Test Case 3	Enter the alphabet: A	The alphabet, A is a vowel.
Test Case 4	Enter the alphabet: B	The alphabet, B is a consonant.

Exercise 5: Read two numbers from the user at the keyboard. Write if-else statements to print the relationship between these two numbers.

Sample Test Cases	**Input**	**Output**
Test Case 1	Enter two numbers : 20, 30	20 <= 30
Test Case 2	Enter two numbers : 40, 30	40 > 30

Extend your program to try the test cases given below.

Sample Test Cases	**Input**	**Output**
Test Case 1	Enter two numbers : 20, 30	20 < 30
Test Case 2	Enter two numbers : 20, 20	20 == 20
Test Case 3	Enter two numbers : 40, 30	40 > 30

Exercise 6: Read the age from the user at the keyboard. Write if-else statements to print if they are eligible to vote. People can vote if they are 18 years old or above. If they are not eligible, tell them how many more years they have to wait to become eligible.

Sample Test Cases	**Input**	**Output**
Test Case 1	*** Voting Eligibility Decider *** Enter your age: 19	You are eligible to vote in the elections.
Test Case 2	*** Voting Eligibility Decider *** Enter your age: 16	Sorry, You are not eligible to vote. You need to wait for 2 more years to become eligible.
Test Case 3	*** Voting Eligibility Decider *** Enter your age: -17	Sorry, age can’t be negative. Please try again.

Exercise 7: Write a program that determines the action based on the student's attendance percentage. Read the total number of classes and the total number of classes attended by the user at the keyboard. Calculate the attendance percentage.
i. If the attendance percentage is >=85%, mentors must write an appreciation note to the student.
ii. If the attendance percentage is between 75% and 85%, mentors must collect an undertaking signed by the parents and the student that the student will be regular hereafter.
iii. If the attendance percentage is < 75%, mentors should call the parents to the department to meet the HoD, Class teacher, and mentor. Collect the undertaking signed by the parents and the student that the student will be regular henceforth.

Sample Test Cases	**Input**	**Output**
Test Case 1	Enter total number of classes : 30 Enter number of classes attended : 27	Attendance percentage is 90. Mentor must write an appreciation letter to the student.
Test Case 2	Enter total number of classes : 30 Enter number of classes attended : 24	Attendance percentage is 80. Mentors must collect an undertaking signed by the parents and the student that the student will be regular hereafter.
Test Case 3	Enter total number of classes : 30 Enter number of classes attended : 20	Attendance percentage is 66.6 Mentors should call the parents to the department to meet the HoD, Class teacher, and mentor. Collect the undertaking signed by the parents and the student that the student will be regular henceforth.
Test Case 4	Enter total number of classes : -30	The total number of classes can't be negative. Try again.
Test Case 5	Enter total number of classes : 30 Enter number of classes attended : -20	The number of attended classes can't be negative. Try again.

Exercise 8: Write a program that calculates the user's body mass index (BMI) and categorizes it as underweight, normal, overweight, or obese, based on the following table

BMI	**Weight Status**
Below 18.5	Underweight

18.5 - 24.9	Normal
25.0 - 29.9	Overweight
30.0 and above	Obese

To calculate BMI based on the weight in kilograms and height in meters, use this formula (rounded to 2 decimal places)

$$BMI = \frac{Weight(kg)}{[Height(m)]^2}$$

Exercise 9: Read three integers from the user at the keyboard. Find the smallest value.

Sample Test Cases	**Input**	**Output**
Test Case 1	Enter three numbers : 34, 15, 27	15 is the smallest number.
Test Case 2	Enter three numbers : 34, -15, 27	-15 is the smallest number.
Test Case 3	Enter three numbers : -4, -27, 0	-27 is the smallest number.

Exercise 10: **WATER BILL PROBLEM PROBLEM:** Write a program that computes a customer's water bill. The bill includes a $35 water demand charge plus a consumption (use) charge of $1.10 for every thousand gallons used. Consumption is figured from meter readings (in thousands of gallons) taken recently and at the end of the previous quarter. If the customer's unpaid balance is greater than zero, a $2 late charge is assessed as well.

Problem Constants

DEMAND_CHG 35.00 /* basic water demand charge */
PER_1000_CHG 1.10 /* charge per thousand gallons used*/
LATE_CHG 2.00 /* surcharge on an unpaid balance */

Problem Inputs

int previous /* meter reading from previous quarter in thousands of gallons */
int current /* meter reading fromcurrent quarter */
double unpaid /* unpaid balance ofprevious bill */

Problem Outputs

double bill /* water bill */
doubleuse_charge /* charge for actual water use */
doublelate_charge /* charge for nonpayment of part ofprevious balance */

Relevant Formulas

water bill = demand charge + use charge + unpaid balance+ applicable late charge

Try the sample test cases given below :

Sample Test Cases	Input	Output
Test Case 1	This program figures a water bill based on the demand charge ($35.00) and a $1.10 per 1000 gallons use charge. A $2.00 surcharge is added to accounts with an unpaid balance. Enter unpaid balance, previous and current meter readings on separate lines after the prompts. Press or after typing each number. Enter unpaid balance> $71.50 Enter previous meter reading> 4198 Enter current meter reading> 4238	Bill includes $2.00 late charge on the unpaid balance of $71.50 Total due = $152.50
Test Case 2	This program figures a water bill based on the demand charge ($35.00) and a $1.10 per 1000 gallons use charge. A $2.00 surcharge is added to accounts with an unpaid balance. Enter unpaid balance, previous and current meter readings on separate lines after the prompts. Press or after typing each number. Enter unpaid balance> $51 Enter the previous meter reading> 4198 Enter current meter reading> 4137	Bill includes $2.00 late charge on unpaid balance of $71.50 Total due = $102.00