

Making Decisions – The Beginnings of AI

Now that your programs have started accepting input, you have begun to notice that it is very easy to cause problems. If someone runs your program and enters a value that is out of range (too big for example) or out of context, you have no way to check, or validate, the value. The solution to this is to learn how to get our programs to make decisions while the program is running, and eventually to have certain commands repeated until a valid response is obtained.

The if() and if() ... else statements

This most commonly used decision making statement is found in a similar form in all programming languages. To be able to make decisions, comparisons must be made to check if a certain condition has been met or has failed. The results of any comparison must be either “YES” or “NO”, which in the program corresponds to the Boolean values of “TRUE” or “FALSE”.

Relational and Logical Operators

Although every programming language has if..else statements, the way that comparisons are performed varies with each language. Just as the mathematical operations are slightly different in C than in ‘regular math’, the comparative operators are also slightly different

Operator Type	Symbol	Purpose
Equality	==	Used to compare if two values are equal
	!=	Used to compare if two values are not equal
Relational	>, >=	Greater than , greater than or equal
	<, <=	Less than, less than or equal
Logical	&&	The AND operator. Used to test if two or more results are TRUE. For multiple checks, it tests from right to left
		The OR operator. Used to test if at least one of two or more conditions is TRUE.

NOTE: One of the most common errors in C programming is the misuse of the == operator. If you use a single =, it assigns a value and then checks it to be true, which it always will be!!!

Examples: *All of the examples are meant to be fragments of code, not complete programs.*

1. Checking a single condition with an if statement

<p>a) to see if a student has paid course fees.</p> <pre>char feesPaid; //lines to input answer ... if(feesPaid == 'y') { print a receipt }</pre>	<p>b) to see if a person is old enough to drive</p> <pre>int age; //lines to input age if(age >= 16) { ...get license }</pre>
--	--

2. Checking a single condition with an if.. else statement.

<p>a) to see if a student has paid course fees.</p> <pre>if(feesPaid == 'y') { print a receipt } else { print a reminder notice }</pre>	<p>b) to see if a person is old enough to drive</p> <pre>if(age >= 16) { ...get license } else { ... tell person how long to wait }</pre>
--	--

3. Checking multiple conditions. Assume correct data types for variables used below.

<p>a) Does a student get a math text book</p> <pre>if((feesPaid == 'y' && inCourse != 'n') { ... issue a book } else { ... issue a notice }</pre>	<p>b) Does student get a CompSci award?</p> <pre>if(mark >= 99 && daysMissed < 5) { ... give an award } else if (mark >90 && daysMissed == 0) { ... give an award }</pre>
---	--

The number of ways that the if.. else can be used is limited only by the need in the problem, and your imagination in how to use it.

Exercises:

1. Determine the result of each of the following:

$8 > 5 \ \&\& \ 3 < 7$		$(3 > 4) \ \ (4 < 5)$	
$(2 + 5) == (4 + 4)$		$(3 + 7) <= (20/2)$	

2. Given the variables and values listed below, determine the result for each expression below.

```
int age = 17;  char passedTest = 'y';    float average = 78.9;
```

a) `if(age >= 16)`

b) `if((age != 16) && (passedTest == 'y'))`

c) `if((average >= 80 && passedTest == 'y'))`

d) `if (age % 2 != 0)`

e) `if(passedTest == 'Y' || age == 16)`

3. Write a C program that prompts the user to enter their age, in years, as an integer, and their gender as a char. After entering these two values, the program will display a message such as “ You are a male who is not yet able to get a driver’s licence”. Note that there are four possible combinations to check for this problem.
4. Write a C program called `gradeMaker.c` that prompts the user to enter the names and marks for four subjects. The program will then display the course names and marks in a neat, tabular form, followed by the average of the marks, and the corresponding letter grade.
5. Write a C program called `WestTrivia.c` that will ask 10 multiple choice questions about life at West Carleton SS (and will allow the user to answer each question). After all ten questions have been answered, have your program output the user’s mark on the quiz, and a message appropriate to their mark. (make a message for 10 out of 10 , 8 or 9 out of 10 etc.)