University of Glasgow

UESTC 1005 - Introductory Programming

Lecture 4 - Operators and Program Control

Dr Hasan T Abbas

About Me

- IP Course Coordinator since 2019
- Research Interests in Numerical Electromagnetics
- Fascinated by Nanoscale Physics

Lecture Outline

- Dive deeper into operators (运算符)
- Make Decisions
- Introduce program control

Operators

In the last lecture, we looked at a number of operators available in C.

C has in fact many more, and mastering all of them takes time.

- Interestingly, the results of operators as sometimes hardware dependent.
- Let's look at some interesting observations.

Some Side Effects

Some unexpected results:

```
int i;
float pi;
pi = i = 3.1416f;
```

The `float` variable `f` is assigned a value of `3.0` (as opposed to `3.1416`).

better to use multiple statements to avoid unexpected behaviour

Example - Operators

Write a C program where you are going to enter a *three-digit* number which is then printed on the screen.

Task is to reverse the number. Example output of the program:

```
Enter a three-digit number: 123
Reversed number is: 321
```

Example - Operators

Method

- We need to extract the three digits from the number
- Split the number `n` into units, tens and hundreds
- Units -- `n % 10` gives us the right-most digit
- Hundreds -- `n % 100` gives us the hundreds
- For tens, we need to do two steps, first remove the hundred `temp = n % 100` and then get the tens part, `ten = temp % 10` gives us the tens in the number

For swapping, we can simply display the number in the reverse order (`unit` , `tens` and `hundred`).

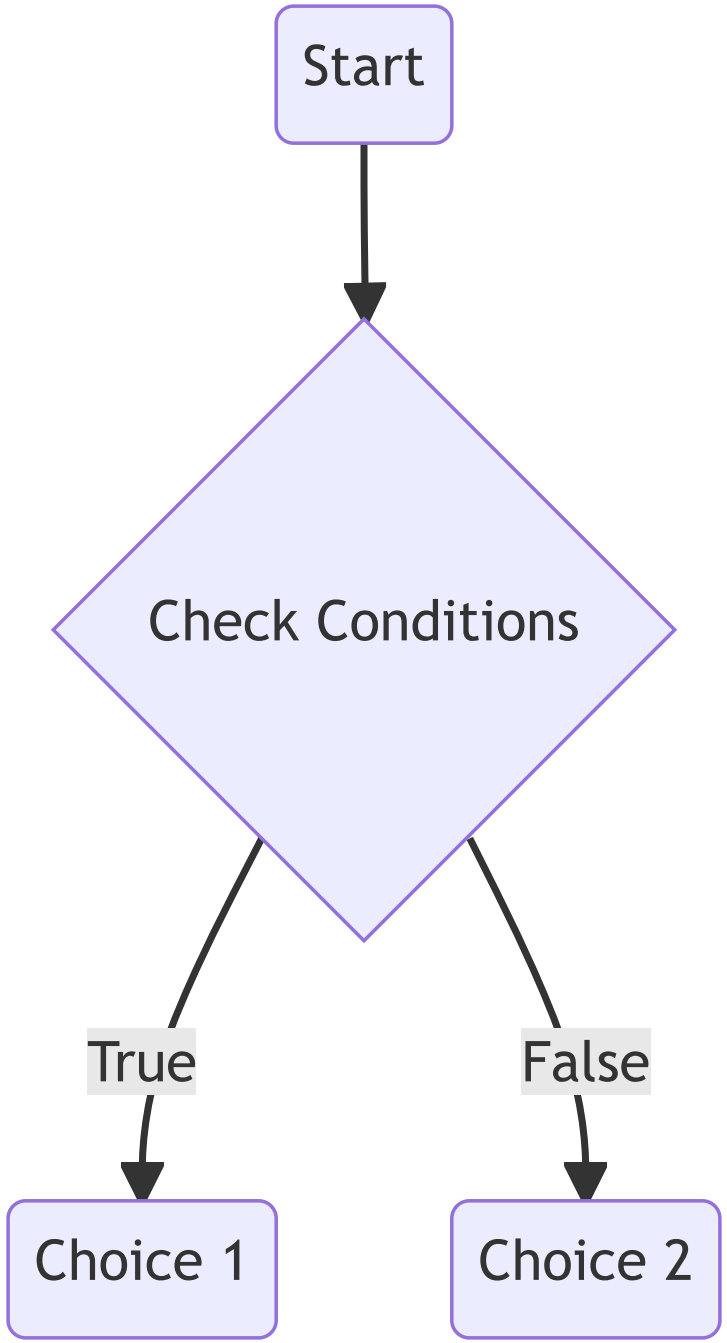
Example - Operators

```c
#include <stdio.h>
int main(void)
{
    int input, unit, ten, hundred;
    int temp; // for temporarily storing a value

    printf("\nEnter a three-digit number: ");
    scanf("%d", &input);

    hundred = input / 100; // Get the hundred
    temp = input % 100;  // remove the hundred
    ten = temp / 10;  // Get the ten
    unit = input % 10; // Get the unit

    printf("\n\nThe reversal is %d%d%d\n\n", unit, ten, hundred);
    return 0;
}
```

Selection and Decisions

C has three kinds of statements (selection, iteration, jump) that can help make decisions and select choices.

Today we will look at `if` and `switch` statements along with relational and logical operators.



Logical Expressions

We need to check an expression to see if it is `true` or `false`.

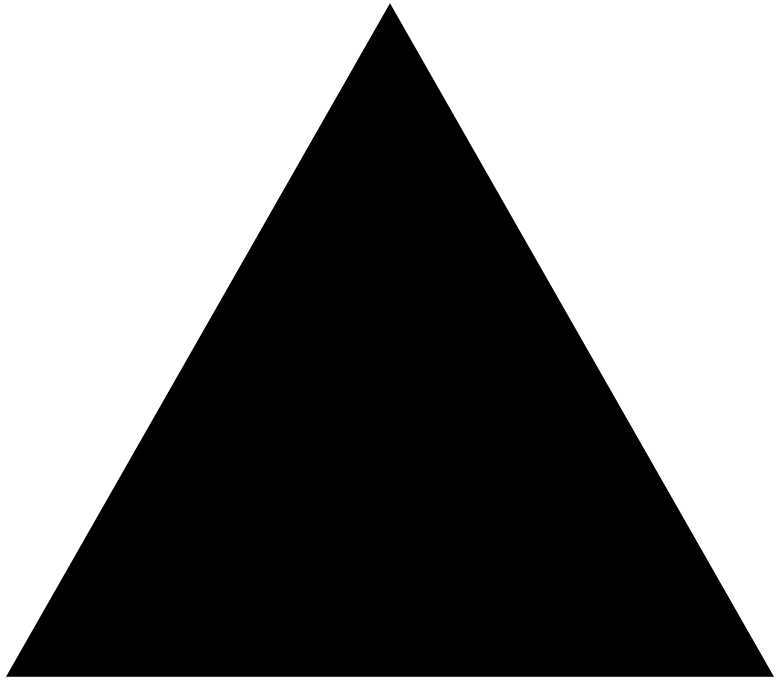
- Like, check whether `i < j` is `ture` or `false`
- The above `(i < j)` is a logical expression.
- Result is an integer, `1` for true, and `0` for false
- we usually call it *Boolean* logic
- Notice we have used *relational operotors* to build logical expressions

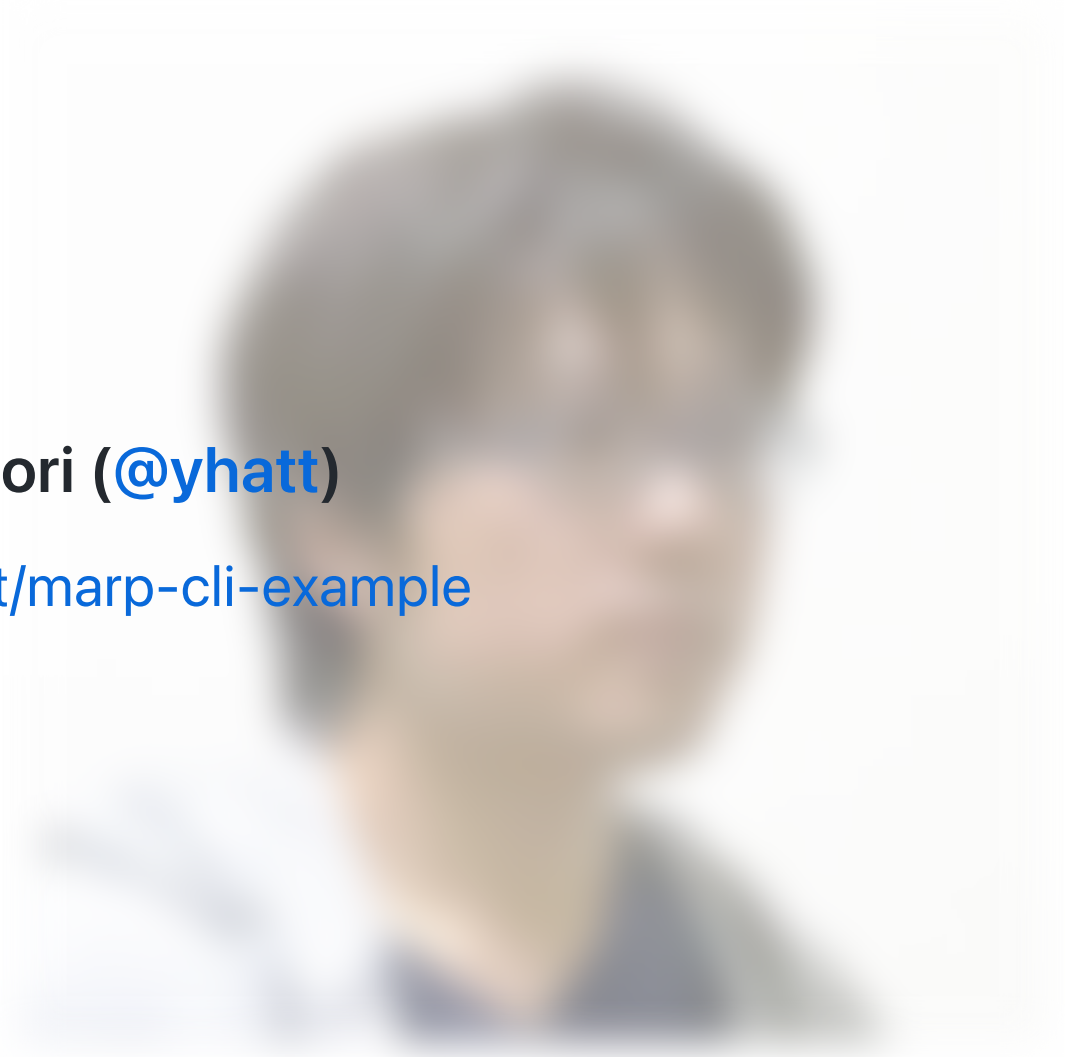
Logical Expressions - Examples

Vercel

Ready to write & host your deck!

▲ Deploy





Created by Yuki Hattori (@yhatt)

https://github.com/yhatt/marp-cli-example