COP 3223H: Introduction to C Programming

Fall 2023

University of Central Florida

Dr. Kevin Moran

Week 5 - Class 1: Grouping & Expressions

Administrivia

- *Small Programming Assignment 2* and *Large Programming Assignment 1* will come out today
  - I will be adjusting the timing of *Small Programming Assignment 3* - moving to later in the semester
- *Quiz 1* is due Wednesday at 11:59 pm
- Heads up on *Exam 1* is this Friday!
  - We will review the format and content extensively in the next class

Today's Agenda

1. Following up on If statements
2. Revisiting Ordering and Grouping of Expressions.

`If` Statements

The `If` Statement

- Conditions are setup in the if statement.
- Syntax example

Condition

Statement Executed if Condition is “true”

```
if(num1 < num2)
    {
        printf("num1 is smaller than num2. \n");
    }else
    {
        printf("num2 is smaller than num1. \n");
    }
```

Statement Executed if Condition is “false”

`If` Statement with One Alternative

- Conditions are setup in the `if` statement.
- Syntax example

```
if(num1 != num2)
        printf("num1 does not equal num2. \n");
```

Multiple Alternative `if-else` Statement

Guess What!! You can include control structures if you have multiple statements that need to be properly executed!!

e”

Nested `if` Statements

- After testing and determining the outcome, it is possible to dive into another condition.
- This is known as creating nested statements.
- Think about nesting dolls!
  - Inside a nest doll is another doll. Inside a nest if statement is another if statement.

```
if (num1 != 0)
    if(num1 !=1)
        if(num1!=2)
            if(num1!=3)
                printf("num is neither 0, 1, 2, or 3 ...");
```

`switch` Statements

`switch` Statement

- Some of the `if else` statements can deal with checking for an exact match.
- What would happen if there are lots of multiple-alternative `if-else` statements that dealt with only equality checks
- Switch Statement allows programmers to write a cleaner version of `if-else` that only deals with == operator.

Q&A: Switch statements use relational operators for comparison?
a) True
b) False

`switch` Statement Syntax

```
switch(ticket)
{
    case 1:
        printf("Proceed to entrance 1.\n");
        break;

    case 2:
        printf("Proceed to entrance 2.\n");
        break;

    case 3:
        printf("Proceed to entrance 3.\n");
        break;

    default:
        printf("Sorry, your ticket does not match!");

}
```

variable being evaluated for equality

ticket == 1

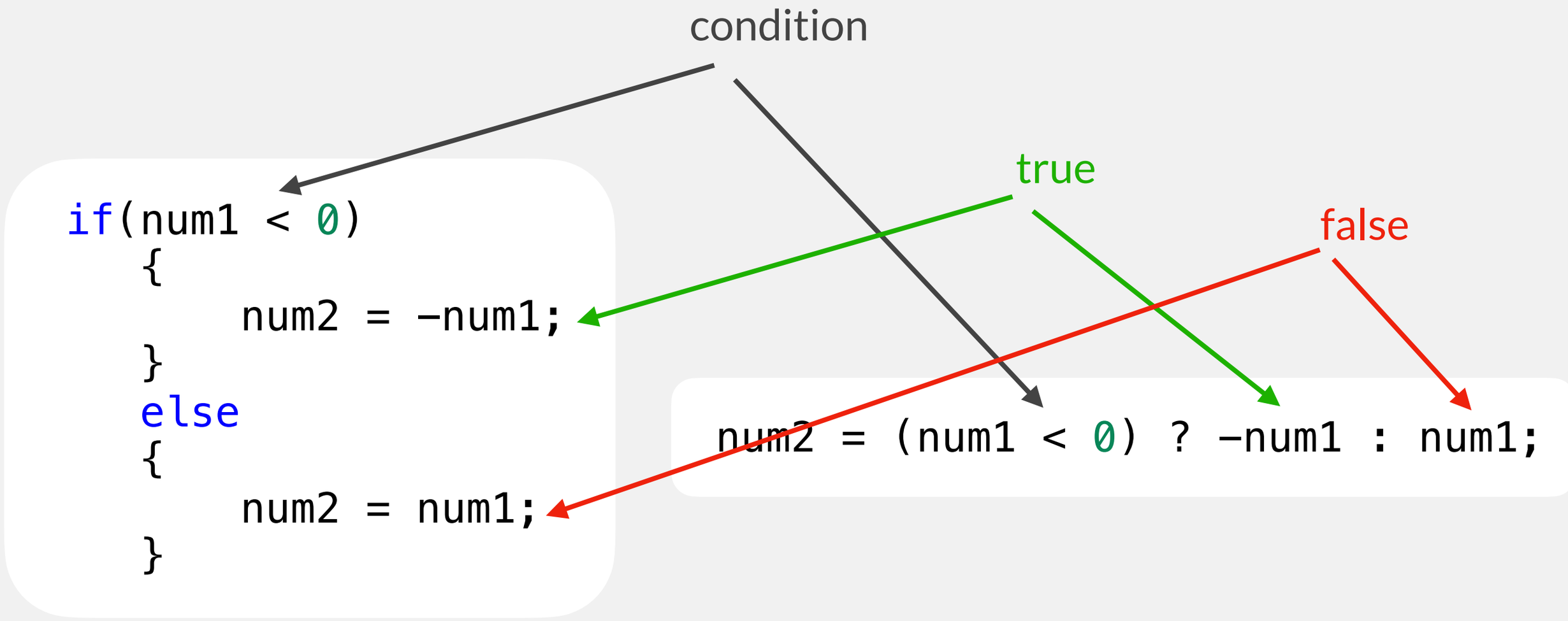
ticket ==2

ticket ==3

ticket !=1 && ticket !=2 && ticket !=3

The Conditional Operator ? :

- C offers a quick simple way to write out an if-else statement in one line of code.



Revisiting Grouping & Expressions

Operator Precedence in C

Operator	Precedence
`function calls`	Highest
`! + - & (unary)`	
`* / %`	
`+ -`	
`< <= >= >`	
`!= ==`	
`&&`	
`||`	
`(=)`	Lowest

Operator Precedence in C

- Precedence determines how operators in C are grouped together.
- When we were writing mathematical expressions in C , we learned that "( )" was how we grouped certain operands together for an operator to perform some sort of action.
- Example:

$$\frac{a+b}{c+d} \rightarrow (a+b)/(c+d)$$

Logical Operator Precedence

- !, &&, || are the 3 logical operators in C we utilize
- A common misconception when we talk about precedence with logical operators is who gets to be executed first.
- *VERY DIFFERENT FROM ORDER OF OPERATIONS!!!*
- When we discuss precedence, we are discussing how logical operators group expressions together and what is being evaluated.

Some Examples

- Assume A, B, C, and D are relation expressions (e.g., `x > y`)
  - `A && B` ⟶ `(A && B)`
  - `A && B || C` ⟶ `((A&&B) || C)`
  - `A || B && C || D` ⟶ `((A || (B&&C)) || D)`
  - `!A` ⟶ `!(A)`

General Advice

- It is good practice to use parenthesis to group operands that you want evaluated by the operator.
- If you don't use parenthesis, then you rely on C's precedence rules!

Acknowledgements

Slides adapted from Dr. Andrew Steinberg's COP 3223H course