Computer Science Principles

Class 2

Github Review

- Merge Pull Requests
- Everybody Sync – up to date
- Questions?

Data Types, Generalized

- Boolean: Must be either TRUE or FALSE
- Integer: Whole numbers, no fraction parts
- Float: Numbers with Fractional parts
- Char: ‘a’ ‘1’, etc. “Something you can type”
- String: “This is a string.” // combination of chars

TRUE or FALSE

- Most languages for boolean variables accept values of TRUE/FALSE.
- Also can be represented by numbers. Usually a 1 == TRUE, and a 0 == FALSE.
- Really, anything that is not zero, is TRUE.

Assignment vs Comparison

- Very common error is to “assign” a variable when you mean to “compare” variables
- Assignment uses “=“ with potentially another operator.
- Comparison uses “==“ or other comparison operators
- Bad example: while (test = TRUE) do something

Comparison Operators

- Equality: ==
- Not equal: !=
- Greater than: >
- Less than: <
- Greater than or equal: >=
- Less than or equal <=

AND/OR Logic

- Focus on logic this class. Some languages can use AND/OR to change variable values
- And: &&   Or: ||
- For an AND(&&) to be TRUE, both sides must be TRUE. If any side is FALSE, then the AND result is False
- For OR to be TRUE, one or more sides must be TRUE. If both sides are FALSE, then the OR will be FALSE.

Negation (NOT)

- The NOT operator ! Reverses the state of whatever it precedes.
- !TRUE is FALSE !FALSE is TRUE
- !(TRUE == TRUE) is FALSE
- !(5 > 6) is TRUE
- We will use the NOT operator as LOGIC not setting values which is possible in some languages

Order of Operations

- Group within parenthesis to ensure comparisons/assignments are what you intend
- (6 > 5) && (TRUE != FALSE) is TRUE
- Multiplication/Division before Add/Subtract

Statement Evaluations (T/F)

- Variable Test = 0; Value = ? Logic = ?

Statement Evaluations (T/F)

- Variable Test = 0; Value = 0 Logic = FALSE
- Test = 3 + 4; Value = ? Logic = ?

Statement Evaluations (T/F)

- Variable Test = 0; Value = 0 Logic = FALSE
- Test = 3 + 4; Value = 7 Logic = TRUE
- Test = TRUE; Value = ? Logic = ?

Statement Evaluations (T/F)

- Variable Test = 0; Value = 0 Logic = FALSE
- Test = 3 + 4; Value = 7 Logic = TRUE
- Test = TRUE; Value = TRUE Logic = TRUE
- Test = !TRUE; Value = ? Logic = ?

Statement Evaluations (T/F)

- Variable Test = 0; Value = 0 Logic = FALSE
- Test = 3 + 4; Value = 7 Logic = TRUE
- Test = TRUE; Value = TRUE Logic = TRUE
- Test = !TRUE; Value = FALSE Logic = FALSE
- Test = TRUE && FALSE; Logic = ?
- Test = TRUE || FALSE; Logic = ?

Statement Evaluations (T/F)

- Variable Test = 0; Value = 0 Logic = FALSE
- Test = 3 + 4; Value = 7 Logic = TRUE
- Test = TRUE; Value = TRUE Logic = TRUE
- Test = !TRUE; Value = FALSE Logic = FALSE
- Test = TRUE && FALSE; Logic = FALSE
- Test = TRUE || FALSE; Logic = TRUE
- Test = (TRUE || FALSE) && !FALSE; Logic = ?

Statement Evaluations (T/F)

- Variable Test = 0; Value = 0 Logic = FALSE
- Test = 3 + 4; Value = 7 Logic = TRUE
- Test = TRUE; Value = TRUE Logic = TRUE
- Test = !TRUE; Value = FALSE Logic = FALSE
- Test = TRUE && FALSE; Logic = FALSE
- Test = TRUE || FALSE; Logic = TRUE
- Test = (TRUE || FALSE) && !FALSE; Logic = TRUE

Coding Guidelines

- Document on Github for reference
- Initially, one file per function. Might expand later.
- Create Stubs first
- Save code that can be used later in “Common code” folder.

Main Function

- Start of any program is main function.
- It will launch and end the program. We will work together to create one

Comments

- Comments are REQUIRED to explain what your code is doing. This helps you and others coming after you understand how your code is working.
- Comment blocks at top of each file, and before each function.