Programming

Michael Brodskiy

Professor: B. O'Connell

September 16, 2022

- Why is Programming Hard?
    - Programming is Detail Oriented
        - Is your syntax right?
            - Did you forget a semi-colon?
            - Is your capitalization correct?
        - Do your instructions make sense?
            - Are they in the right order?
            - Does that input exist yet?
    - There is no room for mistakes
        - You will make mistakes
        - Learn from them though
    - A computer will do exactly what you tell it to
        - Nothing more, nothing less
        - EXACTLY what the code says
    - They do not understand ambiguity
        - They will never infer anything
        - They can't read minds
    - They do not make mistakes
- Algorithmic Thinking
    - Creating instructions for a computer
    - A computer program is an algorithm written using the syntax of a programming language
    - Algorithms are self-contained, step-by-step sets of operations to be performed

- Algorithms
  - A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer
- Algorithmic Efficiency
  - Not all algorithms are created equal
  - The most efficient way may not be the most obvious
  - In this class, though, do not consider efficiency when writing programs
    - Stick with the most obvious algorithm
- Pseudocode
  - An informal version of a program
  - Acts like an outline for your formal code
    - Used to help create formal code
    - Typically becomes comments
  - Steps are in sequence, like in code
- The 5 Steps to Problem Solving
  - State problem clearly
  - Identify inputs/outputs
  - Write a flowchart or pseudocode
  - Write code based on flowchart or pseudocode
  - Test program with a variety of data
    - Learn from mistakes and repeat if necessary
- Programming Concepts
  - Data Handling
    - Value
      - A representation of a specific data type
    - Variable
      - The storage location of that information
      - Typically limits and defines the data type allowed
    - Data Types
      - Integers — Whole numbers, limited by the space set aside (signed or unsigned)

      - Floating Points — Called “floats”, have decimal places
      - Boolean — True or false
      - Characters — Single characters, denoted by single quotes (every character has a corresponding number representation)
      - Data Grouping
        1. Scalar — A single data unit
        2. Array — A unidimensional row or column of data, also known as a vector
        3. Matrix — A two-dimensional grouping of data
      - Strings — Multiple characters, denoted by double quotes (an array of characters)
      - Space Allocation — Size Matters
      - Bit — Binary Digit (0 or 1)
      - Byte — 8 Bits (all data is composed of bytes)
  - Operators
    - Arithmetic
      - +, -, *, /
      - $\wedge$, !
    - Trigonometry
    - Logarithmic
    - “=” refers to assignment, “==” refers to equality
  - Control Structures
    - Branching
      - IFTTT — If, If/Else, If/Else If/Else; Action occurs dependent on a conditional statement or combination of conditionals
    - Looping
      - While — Executes a block of code while a condition is true; checks for condition, if true, runs a block of code until condition is not true
      - Do/While — Executes a block of code, then checks if a condition is true before repeating; runs a block of code, then checks for condition, repeats until condition is not true
      - For — Executes a block of code repeatedly, using a variable with a predetermined set of values to cycle through; starts the loop with a variable (set to initial value), runs the block of code with that variable and value, runs the block of code with that variable and value, repeats until no more values are available, or a condition regarding the values is no longer true

- Data Structures
- Syntax
- Tools