Unit 3: Study Guide

You should be able to:

- Compare natural & programming languages in general terms and justify (explain) the significant differences.
- Trace simple programs & functions to:
  - Predict output
  - Correct errors (add or remove lines of code)
- Work with functions:
  - How to define & call (recognize the code for each)
  - Choosing good names for functions & their parameters
  - Functions can call other functions
  - Function parameters
    - Why creating parameterized functions is valuable
    - How to define a function with parameters
    - How to call a function with parameters
- Loops:
  - How to write a loop
  - Predict what will happen when the loop is executed
  - Predict how many times the loop will iterate
- Loops vs. Functions
  - When to use loops (when is a loop sufficient to accomplish your goal)
  - When to use functions (when is a function required, or at least strongly suggested)
- Abstraction
  - How functions enable higher levels of abstraction
  - How does abstraction enable programming teams to work together more effectively
- Top down design (TDD):
  - Describe the process
  - Provide examples of TDD, given a simple problem
  - Explain how TDD helps programers organize their work and develop abstractions.