Assignment 1

Part 1

1. a) Control flow is an explicit sequence of commands.
   b) Imperative programming organized around hierarchies of nested procedure calls.
   c) Computation proceeds by (nested) function calls that avoid any global state mutation and through the definition of function composition.

   The procedural paradigm improves over the imperative paradigm by adding layers of abstraction in the form of procedures. Procedures interact with contracts, and can wrap local variables.

   The functional paradigm improves over the procedural paradigm by discouraging the use of shared state and mutation, which makes testing, formal verification, and concurrency easier.

   (all definitions were taken from class presentations)

2. $const$ averageGradesOver60 $= (num: number[\,]): number =>$
   $num.filter\big((x: number) => \ x \geq 60\big).map($
   $(x: number, filtered: number[\,]) => x/filtered.length)$
   $.reduce(+,0)$

3. $a)\ (x: T[], y: (z: T) => boolean) => boolean$
   $b)\ (x => number[]) => number$
   $c)\ (x: boolean, y: T[]) => T$
   $d)\ (f: (w: T1) => T2, g: (z: number) => T1) => T2$

4. Abstraction barriers isolate different "levels" of the system. The implementer of the high-level system need not know about low-level details.