PPL - HW1

Barak Gorodissky, Dan Lurie

March 2021

Question 1

(a) The imperative programming paradigm is a paradigm in which the programmer states how commands are sequentially executed and how the state of the program is changing (In cotrast to declarative programming, which states what should be performed and does not specify how).

(b) The procedural programming paradigm introduces the idea of grouping a series of (imperative) commands into procedures/subtourines which can be executed at any point during the program execution. A procedure can call other procedures (or itself), and the program is executed in specific order.

(c) The functional programming paradigm is a paradigm in which each expression/function/composition of functions is evaluated, while not changing the state of the program and having no side effects. It always return the same output for a given input. It is similar to mathematical evaluations of expressions.

- The procedural paradigm give us modularity, using scopes. We do not have to repeat segments of code and the code is much more readable, in contrast to imperative programming where we can not act on blocks of code, and use `goto` instead.
- Fuctional programming does not mutate the state of a program, making it easier to debug and causing less bugs in general. As the state does not change, we can write concurrent programs with less effort, as we only calculate and do not alter the state of the program.

Question 2

(a) `<T>(x: T[], y: (elem: T) => boolean) => boolean`

(b) `(x: number[]) => number`

(c) `(x: boolean, y: T[]) => T`

Question 3

The concept of abstraction barriers is implementing a function using more "basic" functions, without having access to or knowing about their implementation. This idea can be extended to more levels - each additional level uses the immediate level below it.
For example, `stack` package can be implemented using `list`, which in turn can be implemented using an `array`.
`stack` methods, for example `pop`, will use `list` methods, which in turn can be implemented using `array` methods. When we use the `stack` functions, we are not aware of the underlying functions and their implementation.