Principles of Programming Languages

202-1-2051

Homework Assignment 1

Part 1: Theoretical Questions

Question 1

(a) i. Imperative Programming: A program consists of a sequence of commands that change the program's state, running sequentially, this is used to specify to the computer how to accomplish a task.

ii. Procedural Programming: The syntax of the language allows declaration of procedures which can be called from anywhere within the program.

iii. Functional Programming: A functional program is an evaluation of stateless expressions, this paradigm avoids changing states and mutating data. A function is purely mathematical, which means that for every input it will always return the same output and without causing any side-effects.

(b) Procedural programming helps by breaking complicated tasks into smaller simpler tasks which are easier to understand and are more modular.

(c) Functional programming helps by having little to no side-effects, thus being easier to test and debug. The code becomes more reliable, predictable and thread-safe.

Question 2

```
type Product = {
   name: string;
   price: number;
   discounted: boolean;
}

const getDiscountedProductAveragePrice = (inventory: Product[]): number =>
   inventory.filter(x => x.discounted).length === 0 ? 0 :
   inventory.filter(x => x.discounted).reduce((x, y) => x + y, 0)
   / inventory.filter(x => x.discounted).length;
```

Question 3

(a) `<T>(x: T[], y: (value: T) => boolean): boolean => x.some(y)`

(b) `(x: number[]): number => x.reduce((acc: number, cur: number): number => acc + cur, 0)`

(c) `<T>(x: boolean, y: T[]): T => x ? y[0] : y[1]`

(d) `<T, U>(f: (y: T) => U, g: (x: number) => T) => ((x: number): U => f(g(x + 1)))`