**Principles of Programming Languages**

Assignments 1

1. Explain the following programming paradigms:
2. Imperative – consists of commands, focuses on describing how a program operates.
3. Procedural – consists of series of computational steps to carried out, any procedure can be called at any time during the program’s execution.
4. Functional – the program is constructed by applying and composing functions

* how procedural improve imperative: allowing to divide the code into functions enables reusability of the same code without writing different copies in different places, also improves readability and easier to maintain.
* how functional improve procedural: The use of high order functions allows for better understanding of the code. Also , since we have guaranteed immutability of variables , Parallel programming is easier to implement.

1. Types of expressions:
2. (x, y) => x.some(y) = (x: T1[ ], y: pred): boolean
3. x => x.reduce((acc, cur) => acc + cur, 0) = x: number[ ] => x.reduce((acc: number, cur: number) => number.
4. (x, y) => x ? y[0] : y[1] = (x: boolean, y: T1[ ]): T1
5. Abstraction barriers – isolate different ‘levels’ of the system, can be seen in collection of procedures only higher-level procedures can call lower-level procedures. We saw that in the example in practical session with the procedure *printCubes* that applies the function *cube*.