1.

(a) Control flow is a sequence of explicit commands.

(b) Imperative programming is a hierarchy of nested procedures.

(c) Computing is performed through (nested) function calls that avoid any state mutation and through the definition of function composition.

As opposed to the imperative paradigm, the procedural paradigm adds layers of abstraction in the form of procedures. Procedures interact through well-defined contracts, and can encapsulate local variables.

2. Function AverageGradeOver60(f, grades:number)

```
{
Const gradesover60 = grades.filter(gradesover60);
Const total = gradesover60.reduce(function(previousvalue, currentvalue){
        Return previousvalue+currentvalue;
        });
        Return total/gradesover60.length;
}

Function gradesover60(grade: number)
{ return grade > 60 ; }
```

3. (a) <T> (x: T[], y: (z: T) => boolean) => boolean

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

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

The abstraction barriers isolate different "levels" of the system, so the implementer of the high-level system does not need to know about details at the low level.