

CAB402 Programming Paradigms

Assignment 1 - Assessment Criteria

Criteria		Standards			
		Poor (25%)	Fair (50%)	Good (75%)	Excellent (100%)
F# Implementation [20 marks]	Meets functional requirements correctly [10 marks]				<ul style="list-style-type: none"> Unit Tests: / 1493
	Clear, Simple and Easy to Understand [4 marks]				<ul style="list-style-type: none"> All function and variable names are carefully chosen Limited use of nested functions Follows “<i>Single Responsibility Principle</i>” – each function is simple and conceptually does only one thing. Avoids unnecessarily complex logic Documented with high level comments
	Follows Preferred Style [4 marks]				<ul style="list-style-type: none"> Extensive use of higher order functions, lambda expressions and pipeline operator Pure Functional/Stateless
	Performance Improved via State [2 marks]				<ul style="list-style-type: none"> The first implementation is pure A second implementation is provided that substantially improves performance.
C# Implementation [6 marks]	Meets functional requirements correctly [2 marks]				<ul style="list-style-type: none"> Unit Tests: / 466
	Clear, Simple and Easy to Understand Source Code [2 marks]				<ul style="list-style-type: none"> All classes, methods and variable names are carefully chosen All methods are short, simple and easy to follow. <i>Do just one thing</i> <i>Don't make me think.</i>
	Follows Object-Oriented Design Principles [2 marks]				<ul style="list-style-type: none"> Classes abstract concepts in the real world that the application is modelling. Sub-typing used only if there is an is-a relationship. Strong encapsulation of data.
Comparison [4 marks]	Appreciates the strengths and weaknesses of each paradigm				<ul style="list-style-type: none"> Insightful discussion demonstrating appreciation of both practical and conceptual/theoretical issues.