Coursework Feedback:
A Functional Program

This document lays out your scores and commentary feedback for the second coursework of CS141 Functional Programming.

Ingenuity	36/40	This program represents an interpreter for a custom programming language based on a C-like syntax. The program itself works well and the interface is simple. I particularly enjoy the feature of the language where "bare expressions" are printed out to the terminal. This is a nice twist and particularly useful for so-called "printf debugging".

Code Quality	25/30	The code is divided into modules appropriately, emphasising separation of concerns. Monadic style is used well throughout - in particular emphasising the use of parsec, both with the lexer and the parser, in line with the guidance of the venerable Mark Karpov. There are a couple of cases where the code could have been simplified. For example, the Eval module contains a lot of obvious repitition. It would have been possible to abstract out some of the common code in order to massively shorten the implementation of `eval`. Appropriate idioms and language extensions have been used to keep the code simple and readable.

Documentation	20/30	The report is thorough, explaining all aspects of the design process. The inclusion of some demonstrative programs helped to underline the functionality and competence displayed in the implementation. Unfortunately, the code itself includes only minimal documentation, and comments are mostly acting like aides-memoires (plus some witty remarks which I did enjoy). Exercising good comment discipline is an important part of programming, especially in a new language - it helps to show off your low-level understanding and design thinking.

General Comment	Overall this is a very interesting and well implemented project which showcases skill with monadic style. Ensure that your documentation is up to scratch and you will be an excellent functional programming engineer. Great work!

Total mark:	81 / 100