Assignment No. 6 Rubric

EECS 368 - Programming Language Paradigms

Due: 11:59 PM, Monday, November 14, 2022

**Student: Cat Nguyen**

**Student ID: 3077463**

# Point Breakdown

|  |  |  |
| --- | --- | --- |
| ***Graded Value*** | ***Points Possible*** | ***Criteria*** |
|  | 2 | Name of the zip file: FirstnameLastname\_Assignment6 (with your first and last name) |
|  | 2 | Name of the Assignment folder within the zip file: FirstnameLastname\_Assignment6 |
|  | 1 | Copy of Rubric 6.docx with your name and ID filled out |
|  |  | **replicate** function |
|  | 4 | Haskell code for function definition. |
|  | 4 | Function definition includes comments that adequately describe the code. |
|  | 3 | Function definition preceded by its types. |
|  | 4 | Screen print of function executing Example correctly. |
|  | 4 | Screen print of function executing Test Case correctly. |
|  |  | **perfects** function |
|  | 4 | Haskell code for function definition. |
|  | 4 | Function definition includes comments that adequately describe the code. |
|  | 3 | Function definition preceded by its types. |
|  | 4 | Screen print of function executing Example correctly. |
|  | 4 | Screen print of function executing Test Case correctly. |
|  |  | **find** function |
|  | 4 | Haskell code for function definition. |
|  | 4 | Function definition includes comments that adequately describe the code. |
|  | 3 | Function definition preceded by its types. |
|  | 4 | Screen print of function executing Example correctly. |
|  | 4 | Screen print of function executing Test Case correctly. |
|  |  | **positions** function |
|  | 4 | Haskell code for function definition. |
|  | 4 | Function definition includes comments that adequately describe the code. |
|  | 3 | Function definition preceded by its types. |
|  | 4 | Screen print of function executing Example correctly. |
|  | 4 | Screen print of function executing Test Case correctly. |
|  |  | **scalarproduct** function |
|  | 4 | Haskell code for function definition. |
|  | 4 | Function definition includes comments that adequately describe the code. |
|  | 3 | Function definition preceded by its types. |
|  | 4 | Screen print of function executing Example correctly. |
|  | 4 | Screen print of function executing Test Case correctly. |
|  | **100 pts** |  |

|  |  |  |
| --- | --- | --- |
| **Rubric for Program Comments** | | |
| **Exceeds Expectations**  **(90-100%)** | **Meets Expectations**  **(80-89%)** | **Unsatisfactory**  **(0-79%)** |
| Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line. | Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line. | Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line. |

Adequate Prologue Comments:

* Name of program contained in the file (e.g., EECS 368 Assignment 6 - replicate)
* Brief description of the program, e.g.:
  + Haskell function for replicate
* Inputs,e.g.,:
  + Number of replications
  + Element to replicate
* Output, e.g.,
  + List of replicated elements
* Author’s full name
* Creation date: The date you first create the file, i.e., the date you write this comment

Adequate comments summarizing major blocks of code and comments on every line:

* Provide comments that explain what each line of code is doing.
* You may comment each line of code (e.g., using --) and/or provide a multi-line comment (e.g., using {- and -}) that explains what a group of lines does.
* Multi-line comments should be detailed enough that it is clear what each line of code is doing.

# Grader Comments