Assignment No. 7 Rubric

EECS 468 - Programming Language Paradigms

Due: 11:59 PM, Monday, November 13, 2023

**Student: Joon Hee Ooten**

**Student ID: 3010373**

# Point Breakdown

| ***Graded Value*** | ***Points Possible*** | ***Criteria*** |
| --- | --- | --- |
|  | 2 | Name of the zip file: FirstnameLastname\_Assignment7 (with your first and last name) Files in other formats (e.g., .tar will not be graded). |
|  | 2 | Name of the Assignment folder within the zip file: FirstnameLastname\_Assignment7 |
|  | 1 | Copy of Rubric 7.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 7 - 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

# All collaborators

# Other sources for the code ChatGPT, stackOverflow, etc.

# 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.

# Each block of code must indicate whether you authored the code, you obtained it from one of the sources listed in the prolog, or one of your collaborators authored the code, or if it was a combination of all of these.

# Collaboration and other sources for code:

# When you collaborate with other students or use other sources for the code (e.g., ChatGPT, stackOverflow):

# Your comments must be significantly different from your collaborators.

# More scrutiny will be applied to grading your comments in particular explaining the code “in your own words”, not the source’s comments (e.g., ChatGPT’s comments).

# Failure to identify collaborators or other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.

# Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.

# Grader Comments