MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG242 Programming Language Concepts – Spring 2021 - Lab Submission 1

Date handed out: 5 April 2021, Monday
Submission due: 6 April 2021, Tuesday 23:55

Please Read This Page Carefully

Submission Rules

1. **You need to write a comment on the first line of your file**, stating that you read the rules specified here and the submission is your own work. **Submissions without this statement will not be graded.**
   Example, "-- I read and accept the submission rules and this is my own work that is done by myself only"

2. As explained in syllabus[1] provided for CNG 242, attempting any academic dishonesty[2], breaking professionalism and ethics[3] rules may result in the following:
   - You might be asked to perform an oral test to confirm your submission.
   - You may receive a "zero" grade for this submission
   - **You may receive "zero" from all future submissions.**
   - You may receive a failing letter grade for the course and this case might be forwarded to the discipline committee.

3. You cannot use someone else's code.

4. You **cannot hire** someone to write the code for you.

5. **You cannot share your code with someone else.**

6. You need to be ready to demonstrate your own work, answer related questions, and have short coding sessions if necessary.

7. You need to submit a **single Haskell file named with your student id** only. For example, **1234567.hs**

8. Function names must be the same as the provided questions.

9. You cannot share this worksheet with any third parties. Upon doing so, any detected action will directly be sent to the disciplinary committee.

10. **You cannot import libraries; you cannot use anything that we did not cover in the first three weeks. Everything you need is in lab worksheet 1,2,3.**

11. You should only submit one solution per question. Your solution might have multiple lines. **Only the functions with the same name will be graded.**

12. You can only get full marks, if your file fully compiles, runs, and generates correct output for all possible cases. **Non-general static solutions will not be graded!**

13. **If you are added to another section as a guest student**, you will still have to submit your submission under the main section you are registered. You can check your registered section by trying to see your grades in ODTUClass. Only submit your file to the section that you can see your grades.

---

[1] Page 3&4 (Course rules, #1,2,3)
[2] Taking unfair advantage in assessment is considered a serious offence by the university, which will take action against any student who contravenes the regulation through negligence or deliberate intent.
[3] For a comprehensive cheating definition, please refer to: https://ncc.metu.edu.tr/res/academic-code-of-ethics . When a breach of the code of ethics occurs (cheating, plagiarism, deception, etc.), the student will be added to the BLACKLIST.

Questions

1. **[0.5 Marks]** Write a Haskell function called "calculate" that takes two numbers and a character. This function will have four different operations. These are, addition (+ or s), subtraction (- or r), multiplication (* or m), division (/ or d), trailing product(x) and trailing sum (.). Trailing sum will calculate sum of all numbers between the first and second number. Using these characters, it will calculate the result of those operations with given numbers. If the user enters another character, the operation should return 0. Please note that trailing sum and trailing product should work same with both ways and same numbers.

   Sample run:
   ```
   *Main> calculate 3 '+' 5
   8.0
   *Main> calculate 3 '-' 4
   -1.0
   *Main> calculate 5 'x' 3        ←——        3*4*5 = 60
   60.0
   *Main> calculate 3 '.' 5        ←——        3+4+5 = 12
   12.0
   ```

2. **[1 Marks]** Write a Haskell function called "evaluate" that uses calculate function to calculate results of given numbers with given operation. The calculation should be done in a nested way.

   Sample run:
   ```
   *Main> evaluate [1,1,1] "+"        ←——        1+(1+(1)) = 3
   5.0
   *Main> evaluate [2,0,3,1] "-"      ←——        2-(0-(3-(1))) = 4
   4.0
   *Main> evaluate [2,4,6] "."        ←——   2.(4.(6)) = 2.(4+5+6) = 2.15 = [2+3..+15] = 119
   119.0
   ```

3. **[0.5 Marks]** Write a Haskell function called "polynomial" to calculate values of "nested polynomial" functions. These "nested polynomial" functions will be given in list format. For example, [1,1,1] will refer to x^2, x, 1 or [1,2,3] will refer to x^2, 2x, 3 or [2,0,3,1] will refer to x^3, $0x^2$, 3x, 1. The "polynomial" function will also take another value for x and a character to evaluate a given polynomial. **Tip:** you can modify your answer in question 2 to implement this question.

   Sample run:
   ```
   *Main> polynomial [1,1,1] 2 "+"    ←——    1*x²+(1*x+(1)) = 7
   7.0
   *Main> polynomial [2,0,3,1] 1 "-"  ←——  2* x³-(0* x²-(3*x-(1))) = 4
   4.0
   *Main> polynomial [2,0,3,1] 2 "-"  ←——  2*x³-(0*x²-(3*x-(1))) = 16-(0-(6-(1))) = 21
   21.0
   ```

Bonus. [0.5 Marks] Modify question 3 and rename it as "bonus" to accept a list of operators between each operation.

Sample run:

*Main> bonus [1,1,1] 2 "+-" ⟵ $x^2 + (x - (1)) = 4 + (2 - (1)) = 5$
5.0
Main> bonus [2,0,3,1] 1 "+-" ⟵ $2* x^3 *(0*x^2+(3*x-(1))) = 4$
4.0
Main> bonus [2,0,3,1] 2 "+/" ⟵ $16 + (0 * (6 / 1)) = 16$
16.0
*Main> bonus [5,2,1,7,2] 1 "+-" ⟵ $5x^4+(2x^3-$ (Rest is 0, missing operators)) = 7
7.0
*Main> bonus [5,2,1,7,2] 1 "+-+"
6.0

Grading

* Even though the questions are connected, your logic will be evaluated. Therefore, if you have a mistake or missing part in question 1, it will not affect your points for question 2 and question 3.

Grading Point Mark	(Out of 2)
Answer is non-general and static (For any question)	0
Q1 – Solution generates correct output for most of the sample runs but not for all	0.25
Q1 – Fully works	0.5
Q2 – Solution generates correct output for sample runs only but not for all	0.25
Q2 – Solution generates correct output for most cases with a few exceptions	0.5
Q2 – Solution generates correct output for all cases except one specific case	0.75
Q2 – Fully works	1
Q3 – Solution generates correct output for sample runs but not for all	0.25
Q3 – Fully works	0.5
Bonus – Solution generates correct output for sample runs but not for all	0.25
Bonus – Fully works	0.5

Negative Points	(Out of 2)
Not following exact function naming (Per question)	-50%
Not following file naming rule (Check submission rules!)	-1
Not following given function structure	-1
Not applying the first rule (Check submission rules!)	-2
Uploading file in wrong section (Check rule 13)	-0.5
Not following rule 10	Up to -2

Appendix – Q1

```
*Main> calculate 3 '+' 5
8.0
*Main> calculate 4 '-' 5
-1.0
*Main> calculate 6 '-' 2
4.0
*Main> calculate 6 's' 2
8.0
*Main> calculate 6 'r' 2
4.0
*Main> calculate 5 '*' 3
15.0
*Main> calculate 5 'm' (-4)
-20.0
*Main> calculate 5 '/' (-4)
-1.25
*Main> calculate 5 'd' 1
5.0
*Main> calculate 5 'x' 3
60.0
*Main> calculate 3 'x' 4
12.0
*Main> calculate 3 '.' 4
7.0
*Main> calculate 4 '.' 4
4.0
*Main> calculate 4 'x' 4
4.0
*Main> calculate 11 'z' 13
0.0
```

Appendix – Q2

```
*Main> evaluate [5,2,1] '*'
10.0
*Main> evaluate [5,2,7] '/'
17.5
*Main> evaluate [5] '+'
5.0
*Main> evaluate [] '+'
0.0
*Main> evaluate [3] '-'
3.0
*Main> evaluate [3,4,5] '-'
4.0
*Main> evaluate [5,7] 'd'
0.7142857142857143
*Main> evaluate [7,5] 'd'
1.4
*Main> evaluate [8,3,1] 'x'
336.0
*Main> evaluate [8,3,1] '.'
21.0
*Main> evaluate [12,20,0,5,7] '+'
44.0
*Main> evaluate [12,20,0,5,7] '*'
0.0
```

Appendix – Q3

```
*Main> polynomial [5,3,7] 3 'x'
Infinity
*Main> polynomial [] 5 'a'
0.0
*Main> polynomial [] 5 'd'
0.0
*Main> polynomial [3] 5 'd'
3.0
*Main> polynomial [3] 5 'k'
3.0
*Main> polynomial [3,4] 5 'k'
0.0
*Main> polynomial [3,4] 5 '+'
19.0
*Main> polynomial [3,4] 5 's'
19.0
*Main> polynomial [2,4] 1 '.'
9.0
*Main> polynomial [2,4] 2 'x'
4.0
*Main> polynomial [2,4] 1 'x'
24.0
*Main> polynomial [2,4,7] 1 '*'
56.0
*Main> polynomial [2,4,7] 1 '/'
3.5
*Main> polynomial [2,4,7] 2 '/'
7.0
*Main> polynomial [2,4,7,12,11,5,0,-5] 2 '-'
105.0
*Main> polynomial [2,4,7,12,11,5,0,5] 2 '-'
95.0
*Main> polynomial [2,4,7,12,11,5,0,0] 2 '-'
100.0
```

Appendix – Bonus

```
*Main> bonus [3] 3 "m"
3.0
*Main> bonus [3] 3 "x"
3.0
*Main> bonus [3,5,1,7] 2 "dr.x"
-3.4285714285714284
*Main> bonus [2] 4 "q"
2.0
*Main> bonus [] 5 "s"
0.0
*Main> bonus [5,2,1,7,2] 1 "+-"
7.0
*Main> bonus [5,2,1,7,2] 1 "+-*"
7.0
*Main> bonus [5,2,1,7,2] 1 "+-+"
6.0
*Main> bonus [5,2,1,7,2] 1 "srsms"
-8.0
*Main> bonus [5,2,1,7,2] 1 "dr.x"
-3.935979560615581e-7
```