CS 2336 – PROJECT 4 – Preventing a Paradox (Standalone Project 2)

Pseudocode Due: 3/25 by 11:59 PM
Project Due: 4/13 by 11:59 PM

Submission and Grading: All programs are to be submitted in eLearning. The pseudocode will be submitted as a Word or PDF document and is not accepted late. Projects submitted after the due date are subject to the late penalties described in the syllabus. Programs must compile and run in NetBeans 8.0.2. Each submitted program will be graded with the rubric provided in eLearning as well as a set of test cases. These test cases will be posted in eLearning after the due date. Each student is responsible for developing sample test cases to ensure the program works as expected. **Type your name and netID in the comments at the top of all files submitted.**

What to Submit: Save all project files in a directory named with the following format: <netID>_Project<number>. This directory should contain all files associated with the project including the project file itself. Zip up the directory into a standard Zip file. Please do not create a RAR or 7-Zip file, just a .zip file.

Objective: Use object-oriented programming to implement and utilize a binary tree

Problem: Doc Brown has gotten himself in a jam. The software he uses to evaluate integrals has become corrupt and he is in need of assistance. Without this software, his quantum physics calculations will be incorrect, possibly creating time paradox of epic proportions! Doc Brown is enlisting your assistance in the matter. Your future may depend on it.

Details: This program should read basic integrals from a file and evaluate them. If the integral is definite, the program should create the anti-derivative and evaluate for the given interval. If the integral is indefinite, then just the anti-derivative should be created.

Classes

- Use good programming practice for classes – proper variable access, mutators and accessors, proper constructors, etc.
- Remember that classes exist to be used by other people. Just because you don't use it in the program doesn't mean it shouldn't be coded and available for others to use in theirs.
- As with previous projects, you are open to design the classes as you see fit with the minimum requirements listed below
- All classes must be of your own design and implementation. **Do not use the pre-defined Java classes.**
- **Requirements**
    - Binary Tree class
        - Root "pointer"
        - All traversals of the tree will be done recursively
            - This includes functions that add, delete, or search the tree
    - Node class
        - Left and right "pointer"

    Expressions

- Consist of simple polynomial terms - the highest degree will be 10
    - If multiple terms include the same exponent, combine those terms

- Exponents will be represented by the ^ character.
- Do not assume that the expression will be in order from highest to lowest exponent.
- All coefficients will be integers.
- The | (pipe) character will be used to represent the integral symbol
- If an integral is definite, there will be a number before and after the | character
  - $\int_a^b x \, dx$ = a|b x dx
  - The endpoints of the interval for the definite integral will be integer values
- The variable will always be 'x' and the integral will always end with `dx`
- There will always be a space before the first term of the integral and before `dx`
  - Do not assume there will be spaces anywhere else in the expression
- If a term has no coefficient, it is assumed to be 1
- **Example Input:**
  - `| 3x^2 + 2x + 1 dx`
  - `1|4 x^-2 +3x+4 dx`
  - `-2|2 x^3 - 4x dx`

User Interface: There will be no user interface for this program since all input will be read from a file.

Input: All expressions will be read from a file (named `integrals.txt`). There is no limit to the number of expressions that can be in the file. Each expression will be on a separate line.

Output:

- All output should go to a file (named `answers.txt`)
- Each anti-derivative will be printed to a separate line
  - Definite integrals will also include the interval and value (see examples below)
    - Values should be to 3 decimal places
- Use the ^ character to represent exponents
- Order the terms from greatest to least exponent
- Fractions should be simplified
  - All fractions will be enclosed in parentheses
- Add '+ C' to the anti-derivative for indefinite integrals
- **Example Output (based on input above):**
  - `x^3 + x^2 + x + C`
  - `(3/2)x^2 + 4x -x^-1, 1|4 = 35.250`
  - `(1/4)x^4 - 2x^2, -2|2 = 0`

EXTRA CREDIT: Add to your program to allow evaluation of indefinite integrals containing trigonometric functions (potential 15 extra points)

- Simple expressions inside the trig functions
  - Nothing more complex than what you see below
- The trig anti-derivatives will be listed at the end of the expression in the order encountered
  - Consider a path of nodes extending from the right pointer of the smallest exponent node
- **Example Input:**
  - `| sin x + cos x dx`

- o | 1 – cos 4x dx
- o | 3x^4 – 6x^2 + 2sin 10x dx
- **Example Output:**
  - o -cos x + sin x + C
  - o x – (1/4)sin 4x + C
  - o (3/5)x^5 – 2x^3 - 20cos 10x + C