CS 280
Fall 2022
Recitation Assignment 8
April 6, 2023

Due Date: Monday, April 10, 2023, 23:59
Total Points: 6

In this recitation assignment, you are given the definition of a class, called *Value*, which represents values of operands in the SPL language to implement its interpreter in Programming Assignment 3. These include values for the two defined types in the language: Numeric (double), and String (string). The objective of defining the *Value* class is to facilitate constructing an interpreter for the SPL language which evaluates expressions and executes statements using C++.
In RA 8, you are required to implement some of the member functions of the *Value* class that apply the numeric or string operators in order to enable testing the class separately as a unit before using it in the construction of the interpreter in PA3. You are required to implement the following member functions:
`operator*(), operator==(), operator^(), Catenate(), SLthan().`

The semantic rules governing the evaluation of expressions in the SPL language are summarized below:

- Numeric Operators
  - □ The binary operations for addition, subtraction, multiplication, division, and exponentiation (+, -, *, and /) are performed upon two numeric operands. If one or both operands are not numeric, an attempt is made to convert the non-numeric operand to a numeric one. Otherwise, there is a semantic error.
  - □ The exponentiation operator is applied upon numeric operands only. No type conversion from a string to numeric is possible.
  - □ Similarly, the numeric relational operators (==, <, >) operate upon two numeric operands. The evaluation of a numeric relational expression, produces either a true or false value.
  - □ The unary sign operators (+ or -) are applied upon one numeric operand.

- String Operators
  - □ The binary operation for string concatenation (. Dot) is performed upon two string operands. If one or both operands are numeric, an attempt is made to convert the operand to a string one. Otherwise, there is a semantic error.
  - □ Similarly, the string relational operators (-eq, -lt, -gt) operate upon two string operands. The evaluation of a string relational expression, produces either a true or false value.
  - □ The binary operation for string repetition (**) operates upon a string operand as the first operand, where the second operand must be a numeric expression of integer value.

Note: It is recommended to implement the other overloaded operators in the Value class and testing them before using the class implementation in the PA 3 interpreter project.

Vocareum Automatic Grading

- A driver program, called "RA8prog.cpp", is provided for testing the implementation on Vocareum. The "RA8prog.cpp" will be propagated to your Work directory, along with the definition of the *Value* class in the "val.h" file.
- You are provided by 5 test case files associated with Recitation Assignment 8. Each test case checks the implementation of one of the overloaded operator functions. Vocareum automatic grading will be based on the produced output of your implementations for the required overloaded operators compared with the test case file. You may use them to check and test your implementation. These are available in a compressed archive "RA 8 Test Cases.zip" on Canvas assignment.
- "RA8prog.cpp" is available with the other assignment material on Canvas. Review the driver program for implementation details.

Submission Guidelines

- Please upload your implementation to Vocareum as a "val.cpp" file. The file should include the implementations of the Value member functions: `operator*()`, `operator==()`, `operator^()`, `Catenate()`, `SLthan()`.
- **Submissions after the due date are accepted with a fixed penalty of 25%. No submission is accepted after Wednesday 11:59 pm, April 12, 2023.**

Grading Table

Item	Points
Compiles Successfully	1
Test case 1: Multiplication operation	1
Test case 2: Numeric equality operation	1
Test case 3: Exponentiation operation	1
Test case 4: Catenation operation	1
Test case 5: String Less Than Operation	1
Total	6