CSCE 240 – Programming Assignment Five

Due: 11:59pm on Thursday, November 17

Purpose – Create the Fraction (Parent) and Probability (Child) classes

Create a *Fraction* class that holds the numerator and denominator in private non-static integer data members. The class will contain a constructor with default arguments, accessor and mutator functions for the private data members, an overloaded == operator, and functions ToDecimal, Multiply, Divide, and ToString. Read the comments in the supplied *fraction.h* header file for more detailed requirements. Initial tests for the functionality of the *Fraction* class have been provided in the attached *TestFraction*.cc* source files. Note: all tests except ToString and ToDecimal use the == operator. You are encouraged to create more rigorous tests.

Derive a *Probability* class from the *Fraction* base class. The class will contain a constructor with default arguments, a constructor to convert a *Fraction* object to a *Probability* object, an OddsInFavor function, and redefined functions from the *Fraction* class as needed. Read the comments in the supplied *probability.h* header file for more detailed requirements. Initial tests for the functionality of the *Probability* class have been provided in the attached *TestProbability*.cc* source files. Note: all tests use the == operator. You are encouraged to create more rigorous tests.

Specifications

- Add all code for the definition of the *Fraction* class to the attached *fraction.h* header file.
- Include all of the necessary code for the *Fraction* class, including the implementation all of the public member functions, in the attached *fraction.cc* source file.
- Add all code for the definition of the *Probability* class to the attached *probability.h* header file.
- Include all of the necessary code for the *Probability* class, including the implementation all of the public member functions, in the attached *probability.cc* source file.
- You will submit a zip file (only a zip file will be accepted) containing *fraction.h, fraction.cc, probability.h* and *probability.cc* to the assignment in Blackboard.
- Source files must compile and run on a computer of the instructor's choosing in the Linux lab (see your course syllabus for additional details).

Grade Breakdown

Style *fraction.h*: 0.25 point

Style *fraction.cc*: 0.25 point

Style *probability.h*: 0.25 point

Style *probability.cc*: 0.25 point

Documentation: 1 point
Clean compilation of *fraction.cc*: 0.4 point
Clean compilation of *probability.cc*: 0.4 point
Clean compile/link with *TestFractionConstructors.cc*: 0.1 point
Clean compile/link with *TestFractionToDecimal.cc*: 0.1 point
Clean compile/link with *TestFractionToString.cc*: 0.1 point
Clean compile/link with *TestFractionMultiply.cc*: 0.1 point
Clean compile/link with *TestFractionMultiplyOperator.cc*: 0.1 point
Clean compile/link with *TestFractionDivide.cc*: 0.1 point
Clean compile/link with *TestProbabilityConstructors.cc*: 0.1 point
Clean compile/link with *TestProbabilityToDecimal.cc*: 0.1 point
Clean compile/link with *TestProbabilityToString.cc*: 0.1 point
Clean compile/link with *TestProbabilityMultiply.cc*: 0.1 point
Clean compile/link with *TestProbabilityMultiplyOperator.cc*: 0.1 point
Clean compile/link with *TestProbabilityDivide.cc*: 0.1 point
Passes instructor's modified *TestFractionConstructors.cc* tests: 0.5 point
Passes instructor's modified *TestFractionToDecimal.cc* tests: 0.5 point
Passes instructor's modified *TestFractionToString.cc* tests: 0.5 point
Passes instructor's modified *TestFractionMultiply.cc* tests: 0.5 point
Passes instructor's modified *TestFractionMultiplyOperator.cc* tests*:* 0.5 point
Passes instructor's modified *TestFractionDivide.cc* tests*:* 0.5 point
Passes instructor's modified *TestProbabilityConstructors.cc* tests: 0.5 point
Passes instructor's modified *TestProbabilityToDecimal.cc* tests: 0.5 point
Passes instructor's modified *TestProbabilityToString.cc* tests: 0.5 point
Passes instructor's modified *TestProbabilityMultiply.cc*t tests: 0.5 point
Passes instructor's modified *TestProbabilityMultiplyOperator.cc* tests*:* 0.5 point
Passes instructor's modified *TestProbabilityDivide.cc* tests: 0.5 point

The penalty for late program submissions is 10% per day, with no submission accepted after 3 days.