**CPP Problem Design**

|  |
| --- |
| **Subject:** **Fraction** |
| **Contributor: 林承達, 陳俊儒, 廖宣瑋** |
| **Main testing concept:** Class   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ C++ BASICS  □ FLOW OF CONTROL  ■ FUNCTION BASICS  □ PARAMETERS AND OVERLOADING  □ ARRAYS  ■ STRUCTURES AND CLASSES  □ CONSTRUCTORS AND OTHER TOOLS  □ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES  □ STRINGS  □ POINTERS AND DYNAMIC ARRAYS | □ SEPARATE COMPILATION AND NAMESPACES  □ STREAMS AND FILE I/O  □ RECURSION  □ INHERITANCE  □ POLYMORPHISM AND VIRTUAL FUNCTIONS  □ TEMPLATES  □ LINKED DATA STRUCTURES  □ EXCEPTION HANDLING  □ STANDARD TEMPLATE LIBRARY  □ PATTERNS AND UML | |
| **Description:**  Define a class for a type called Fraction. This class is used to represent a ratio of two integers. Include mutator functions that allow the user to set the numerator and the denominator. Also include a member function that returns the value of the numerator divided by the denominator as a double, **but if the value can represent by an integer, then it should be represented integer**. Include an additional member function that outputs the value of the fraction reduced to lowest terms. For example, instead of outputting 20/60 the function should output 1/3. This will require finding the greatest common divisor for the numerator and denominator, and then dividing both by that number. Embed your class in a test program.  **Input:**  No input.  **Output:**  As following sample.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | #include “Fraction.h”  int main()  {  Fraction f1, f2;  f1.setNumerator(4);  f1.setDenominator(2);  f1.getDouble();  f1.outputReducedFraction();  f2.setNumerator(20);  f2.setDenominator(60);  f2.getDouble();  f2.outputReducedFraction();  return 0;  } | 2  2  0.333333  1/3 | |
| **■ Eazy,Only basic programming syntax and structure are required.**  **□ Medium,Multiple programming grammars and structures are required.**  **□ Hard,Need to use multiple program structures or more complex data types.** |
| **Expected solving time:**  15 minutes |
| **Other notes:** |