

I. Matching: choose the correct fittings (10 %)

- | | |
|---|---|
| <p><u>h</u> 1. class string</p> <p><u>g</u> 3. Data abstraction</p> <p><u>h</u> 5. Copy constructor</p> <p><u>b</u> 7. Iterator</p> <p><u>f</u> 9. Dangling pointer</p> | <p><u>i</u> 2. Memberwise assignment</p> <p><u>e</u> 4. static class variable</p> <p><u>e</u> 6. friend function</p> <p><u>e</u> 8. Pointer-based arrays</p> <p><u>a</u> 10. this pointer</p> |
|---|---|

- a) An implicit argument to all non-static member-function calls.
- b) An object that "walks through" a collection.
- c) Defined outside the class's scope, yet has access to private members of the class.
- d) Used when only one copy of a variable should be shared by all instances of a class.
- e) Do not provide range-checking.
- f) Problem that may occur when a pointed object is out of the scope.
- g) A constructor that takes as its argument a reference to an object of the same class as the one in which the constructor is defined.
- h) Describing functionality of a class independent of its implementation.
- i) The default behavior of the = operator.
- j) Provides member function substr.

II. (Rational Class) As you did in the exercise, create a **Rational** class for storing fractions in arithmetic. Use two private integer variables **int numerator** and **int denominator** to hold the two parts of a fraction. Provide a constructor that initialize an object of the class when it is declared. For example, the fraction

$$\frac{2}{4}$$

would be stored as 2 in the **numerator** and 4 in the **denominator** in the object. Create public **member functions** that perform each of the following tasks:

- a) A default constructor that uses *member initializer syntax* to set the data members of an object. This constructor should also use default arguments of values 1/1. (10%)
- b) A copy constructor that sets the data members of a newly created object to the data members of the **Rational** object argument. (5%)
- c) An **addRational** member function that adds another Rational object to an object itself. Use this pointer in this member function. (10%)
- d) A **printRational** member function that output a **Rational** object in the form **a/b**, where **a** is the numerator and **b** is the denominator. (5%)
- e) A **getRational** member function that input two integers for the numerator and the denominator data members of a Rational object. (5%)

- f) Overload the **addition operator (+)** to add another Rational object to an object itself. Use this pointer in this member function. (10%)
- g) Overload the stream insertion operator **<<** to output a Rational object in the form **a/b**, where **a** is the numerator and **b** is the denominator. (10 %)
- h) Overload the **++** and the **-- operators** for **pre- and post- operations** that adds 1 to and minus 1 from a Rational object. (10%)

III. Again, create a **Rational** class for storing fractions in arithmetic. This time use a private **C structure** data member that integrates two integer variables **int numerator** and **int denominator** to hold the two parts of a fraction. (25%, a:5, b:10, c:10)

- a) Please create a C structure **Rational** with two integer variable fields for the **numerator** and **denominator** of a fraction.
- b) Please create a class **RationalClass** that has a data member of **Rational structure**. Define a constructor that accepts two arguments, e.g. 3 and 4 and uses *member initializer syntax* to set the data fields of the fields of the structure data member.
- c) Overload the **multiply operator (*)** to multiply two Rational objects and returns the result object.

$$\frac{1}{3} \times \frac{1}{4} = \frac{1 \times 1}{3 \times 4} = \frac{1}{12}$$

IV. 一位科學家和浪漫的詩人同乘一輛火車，雖然兩人互不相識，卻挺有話聊的。科學家對詩人說：「你要不要玩個遊戲？我們互相問對方問題，答不出來的要給對方10元，如何？」

詩人心想自己這麼窮酸，比賽又不太容易贏科學家，於是就推託拒絕了。好勝的科學家依然不死心的說：「那這樣子好了，如果是我答不出，就輸給你100元，這樣好不好？」在金錢的誘惑下，詩人也就答應了科學家。

科學家問：「地球表面距離地心有幾公里？」詩人答不出來，就拿了10元給科學家。

接著詩人就問：「什麼東西上山時是四條腿，下山時是五條腿？」科學家這下子可是被考倒的，就這樣一直到了火車到站時，他還是答不出答案是什麼，於是他就心甘情願拿了100元給那詩人。

科學家不解的問：「告訴我答案是什麼？快告訴我吧！」只見詩人默默不語，從口袋掏了10元給他。

結論：考題寫不出來要付出代價的！