1. 9.2 p. 350:

Constructors are a special kind of function, with three peculiarities:

* Constructors must have the same name as the class itself.
* Constructors do not have a return type – not even void.
* Constructors are invoked when an object is created. Constructors play the role of initializing objects.

1. 9.10 p. 354:

The class definition and implementation may be in two separate files. Both files should have the same name but different extension names. The class definition file has an extension name .h (h means header) and the class implementation file has an extension name .cpp.

1. 9.17p. 359:

The benefits are to prevent data from being tampered with and to make the class easy to maintain.

1. 10.15 p. 389:

An instance variable is tied to a specific instance of the class; it is not shared among objects of the same class.

Static variables store values for the variables in a common memory location. All objects of the same class are affected if one object changes the value of a static variable.

1. 10.17 p. 389:

c.f() - yes

C::f() - yes

c::f() – no

1. 10.20 p. 392:

Inconsistent use of const.

Either change getNumber to a const member function or remove const in the printA header.

1. 11.30 p. 435:

Line 8 creates an anonymous object dynamically on the heap, which causes memory leak. You should not create an anonymous object using the new operator.

1. 14.4 p. 534:

In C++, Lvalue (short for left value) refers to anything that can appear on the left side of the assignment operator (=) and Rvalue (short for right value) refers to anything that can appear on the right side of the assignment operator (=).

1. 14.12 p. 536:

It is correct. The implementation in the book is better because it adds denominator directly to numerator rather than invoking the add function.

1. 14.14 p. 538:

You type the keyword “friend” before the class prototype. Then, you can just use dot operator to access a class’s private members.