

Student: Labarbera, Sterling (spl2q2)

Practical Problem: RAI

Score: 11 / 15 points

Grader's feedback on the source code you submitted for this problem

---

- Line 22. (-1 point) As a general rule, a class's data members should not be declared 'public'. A class's data members should generally be declared 'private' (preferred) or 'protected'. For some class X, you want to ensure the data members inside a class X object are accessible only within class X's own methods. This is done by declaring class X's data members as private or protected (i.e., ensuring X's data members are not public).

```
private:
    int id_ ;
```

- See also: C++ data encapsulation; C++ information hiding; C++ data hiding
- Line 22. (-1 point) The global function **device\_open()** should not be invoked within class **device\_guard**. Also, the assignment statement on line 22 is not allowed in C++.
  - NB: Function **main()** invokes **device\_open()** and passes the returned ID value to the **device\_guard(int)** constructor method to instantiate the **device\_guard** object **dg** (see line 54).
- Lines 24 – 26. With constructor methods, prefer to use a [data member initializer list](#) to assign initial values to a class object's data members. Do this instead of assigning the initial values inside the body of the constructor method.

```
device_guard(int id)
: id_{id} // data member initializer list
{ /* empty */ }
```

- Lines 31 – 33. (-2 points) The access specifier 'public' requires a trailing colon ':', and each access specifier keyword is typically placed on a line by itself. Also, class method **device\_id()** should have a return type of 'int'.

```
public:
    int device_id() { return id_; }
```

- Good job on your implementation of RAI! You correctly used a **device\_guard** class constructor to save the acquired resource (the device ID value) within the **device\_guard** object **dg** (see line 54 in function **main()**), and when **dg**'s lifetime ends, class **device\_guard**'s destructor method correctly calls global function **device\_close()** and passes it the ID value that is saved within object **dg** (effectively, **device\_close(dg.id\_)**).