Pg. 803-807 / 2-16 even, 44, 46, 48, 74, 76

Pg. 895 / 6, 8, 10

2.) The name and age variables default to private within the class, but are public in the structure.

4.) getRadius is of the Circle class

6.) mutator function change the value of a classes variables while accessors extract the data from the class

8.) It requires std namespace, at least one more include in the class, and it may be harder to track the use of such statements potentially resulting in lost characters or input.

10.) A constructor is called when a class instance is created. A destructor is called automatically when an instance of the class falls out of scope.

12.) More than one constructor is possible but each must have unique parameters. Only on destructor is allowed per class and may not have any parameters.

14.) Either an initializer list or manually initialize each element using a constructor call.

16.) List nouns and their commonalities is a good way to narrow down what would be best as a class.

44.) Circle::Cirlce(int r = 0) { radius = r; }

46.) Circle circles[5];

48.) for(int a = 0; a < 5; a++) {

cout << “Circle “ << a << “:” << endl;

cout << “Radius: “ << circles[a].getRadius() << endl;

cout << “Area: “ << circles[a].getArea() << endl;

}

74.) The “class Moon” should not be followed by a semicolon. This semicolon should be moved to the very end of the class definition, after the closing curly bracket. “Class”, “Private”, and “Public” should be lowercase. “Public” and “Private” appear to have semicolons, when they should have colons. Member functions defined in line should not have semicolons before the curly brackets. moonWeight() looks like a constructor but is improperly named. It would need to be named “Moon”. In main, the first cout and cin use the wrong extraction and insertion operators. When lunar is initalized, there is no constructor available with matching parameters. If moonWeight was fixed, then this would work.

76.) The Change constructor is missing some parameter types for d and q in both the prototype and the function definition.

**Used 8th Edition book for the chapter 14 questions**

6.) It is the use of a copy constructor. This involves copying the variables from one instance of a class to a new instance of the same class. Pointers can make this tricky.

8.) A copy constructor is a constructor that only takes an instance of the same class, passed by reference, as a parameter.

10.) If it is not an infinite loop attempting to copy the class will result.