D – Testing and Debugging

Student Exercises – Documentation

# Exercises Overview

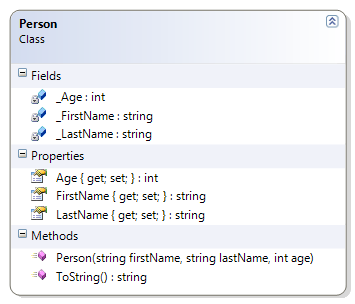
The following exercises are given as practice for this topic. The recommended order of the exercises is as follows.

1. **Person** - This simple class was used to introduce the idea of encapsulation (private fields with public properties). This example uses a constructor for ensuring the state of an object when it is instantiated (created). It also demonstrates the overriding of the ToString() method inherited from the Object method.
2. **CanadianAddress** - This class represents an address for some place in Canada.  
   Update this class to now include a constructor and to override the ToString() method.
3. **Course** - This class represents a post-secondary course with a theory (exam) and a lab portion.
4. **ExamResult** - This class represents the results of an exam for a student.
5. **LabResult** - This class represents the results of a lab for a student.

For this exercise, take the code solutions you created in the previous topic and compare your results with the jUnit tests for these classes; check with your instructor to see where you can download the unit tests.

#### Person

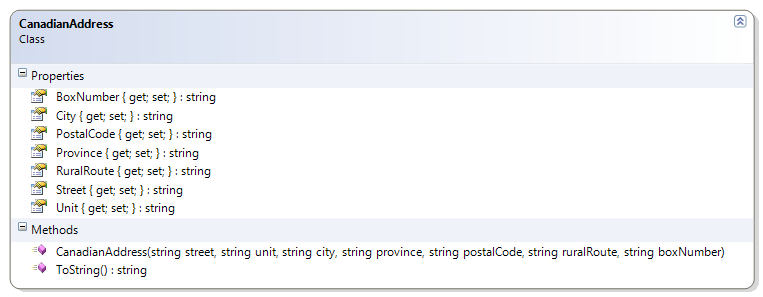
This simple class was used to introduce the idea of encapsulation (private fields with public properties). This example uses a constructor for ensuring the state of an object when it is instantiated (created). It also demonstrates the overriding of the ToString() method inherited from the Object method.



Should…

* Instantiate (build) from constructor
* Get/Set First Name
* Get/Set Last Name
* Get/Set Age
* Override ToString() to get the person’s full name (as first name then last name)

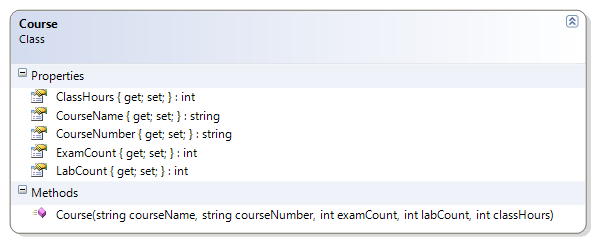
#### CanadianAddress



Should…

* Instantiate (build) from Constructor
* Get/Set: Street, Unit, City, Province, Postal Code, Rural Route, Box Number
* Override ToString() to just show Street, City, Province, and Postal Code

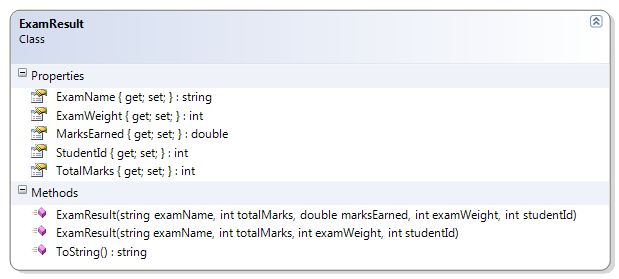
#### Course



Should…

* Instantiate from Constructor
* Get: Course Name, Course Number, Exam Count, Lab Count, Class Hours

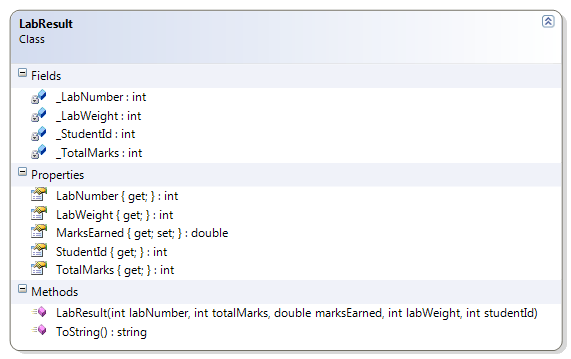
#### ExamResult



Should…

* Instantiate from Constructor
* Get/Set: Marks Earned
* Get: Name, Student Id, Total Marks, Exam Weight
* Override ToString() to show   
  “The student (*studentId*) received *earnedMarks*/*totalMarks* for this *examName* exam.”

#### LabResult



Should…

* Instantiate from Constructor
* Get/Set: Marks Earned
* Get: Lab Number, Student Id, Total Marks, Lab Weight
* Override ToString() to show   
  “The student (*studentId*) received *earnedMarks*/*totalMarks* for this lab.”