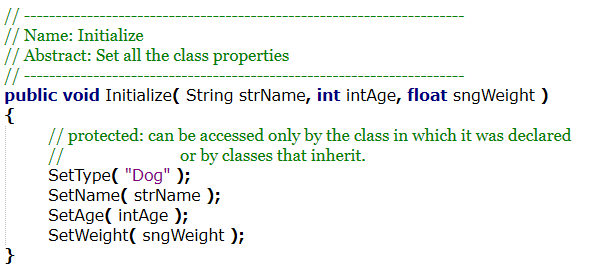
**IT-161 – Java Programming #1**

#### Homework Constructors

**Step 1 – Constructors and overloading**

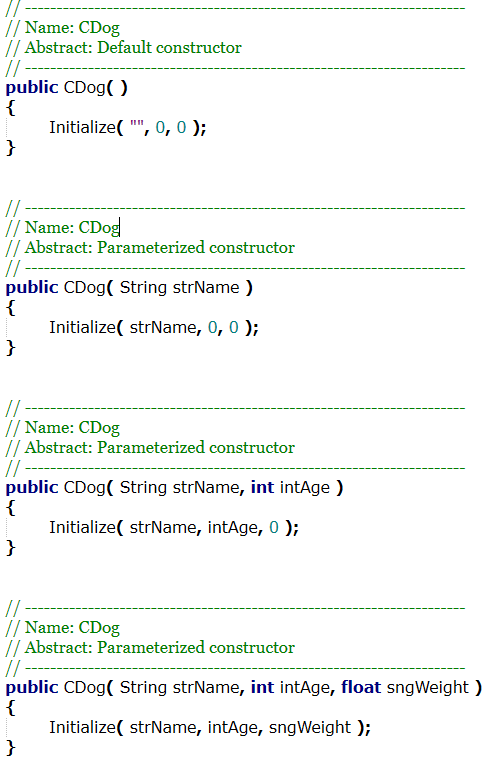
1. Use your last homework assignment as a starter file.
2. Add an initialize method to your CDog class and call the set methods from inside the procedure.

* public void Initialize( String strName, int intAge, float sngWeight )

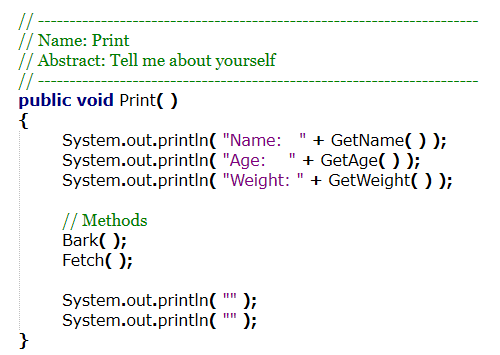


1. Add the following default constructor and parameterized constructors. Call the Initialize method from the previous step inside the constructors.

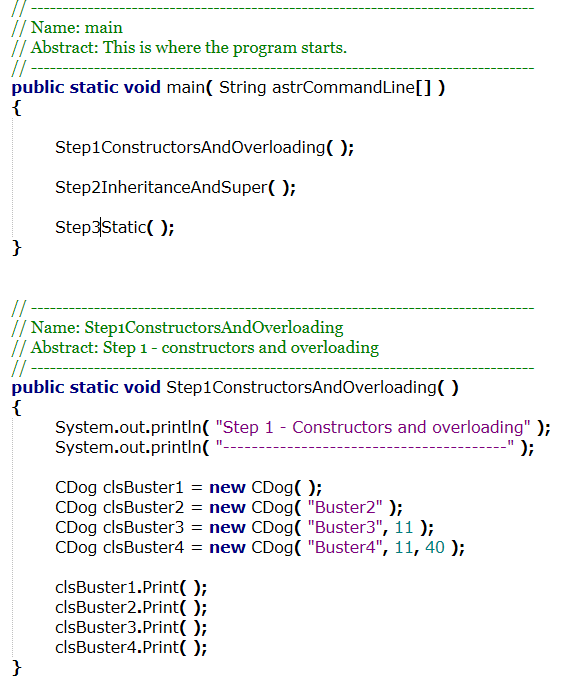
* public CDog( )
* public CDog( String strName )
* public CDog( String strName, int intAge )
* public CDog( String strName, int intAge, float sngWeight )

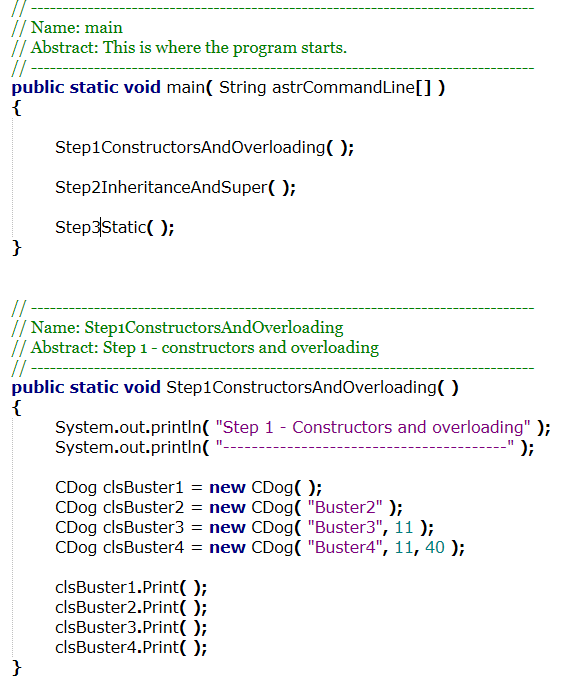


1. Add a Print method to your CDog class that prints out all the properties in the class and calls each method.



1. Make a Step1 procedure in HWConstructors.java and call it from main. In the Step1 procedure make four different instances of the CDog class each using a different constructor. Call the Print method from the previous step to show the parameterized constructors correctly initialized each instance.



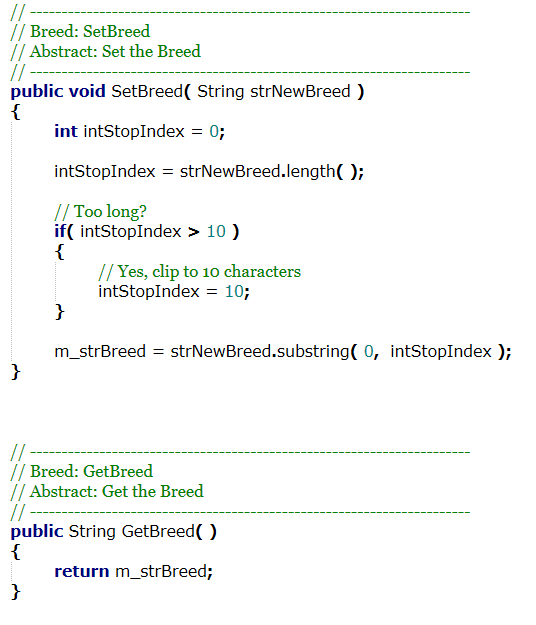


1. You have now successfully implemented constructors and overloading.

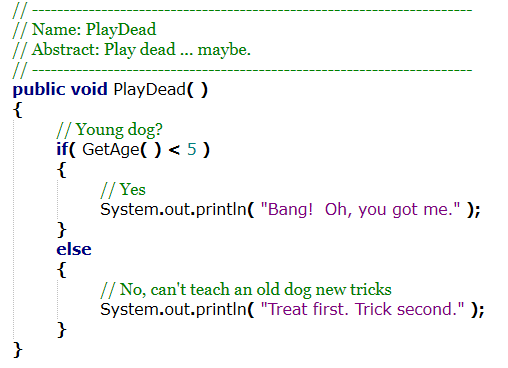
**Step 2 – Inheritance and super**

1. Create a derived class CTrainedDog that inherits the base class CDog.

* Add at least one new private property (e.g. Breed) with public get/set methods. Include some sort of boundary checking in the set method.

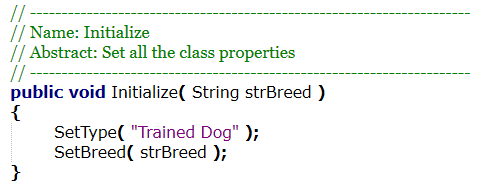


* Add at least two new methods (e.g. PlayDead). Include some sort of centralized business logic that uses one or more of the properties in either the CDog class or the CTrainedDog class to produce different results.



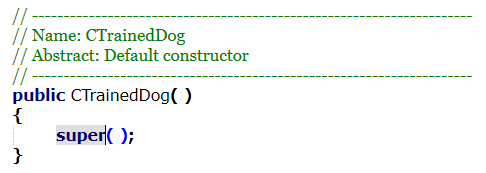
1. Add an initialize method to your CTrainedDog class that takes a value for the new property you created in the previous step. Call the set methods from inside the procedure.

* public void Initialize( <new property value> )



1. Add the following constructors:

* Add a default constructor. Call super( ) with no parameters to call the default constructor from the parent CDog class.



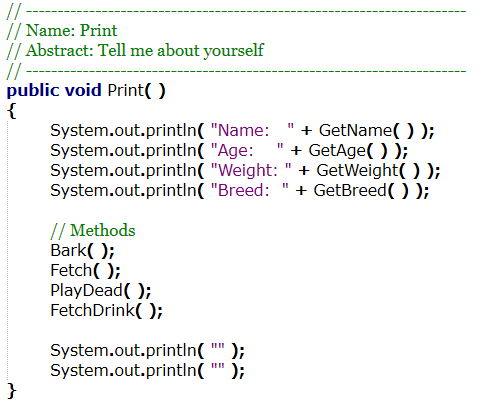
* Add three parameterized constructors that mirror the three parameterized constructors in CDog. Call the super( ) method with the appropriate parameters.



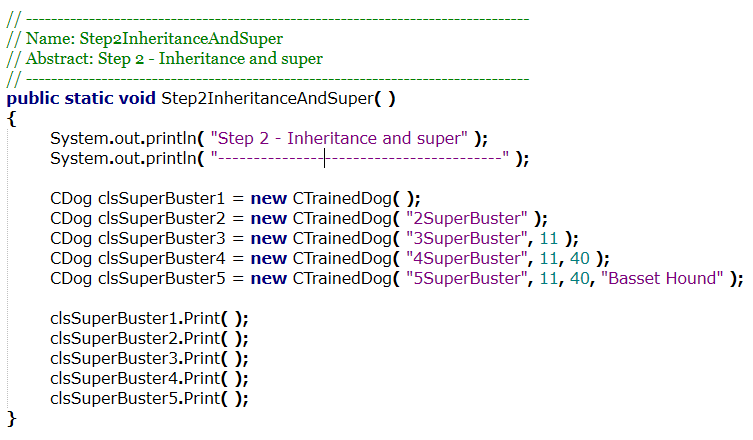
* Add one parameterized constructors that take four parameters. It should take name, age, weight and a value for the new trained dog property. Call super( ) with the correct parameters and then call the initialize method with the trained dog property.



1. Add a Print method to your CTrainedDog class that prints out all the properties in the class and calls each method.



1. Make a Step2 procedure in HWConstructor.java and call it from main. In the Step2 procedure make five different instances of the CTrainedDog class each using a different constructor. Call the Print method from the previous step to show the parameterized constructors correctly initialized each instance.



1. You have now successfully implemented inheritance and super.