**Instructions**

**You will create two classes to work together: House.java and HouseException.java**

This assignment will help students...

* remember their ICS111 object-creation and validation skills.
* get familiar with how to write and submit assignments in ICS211

**House.java**(160 points total)

*Instance variables and validity restrictions*

   private int mls; integer ranges between 10001 and 99999  (MLS is a unique listing number for a home when on sale)  
   private int bedrooms; integer ranging from 0 to 5  
   private double price; double ranging from 0 to 1,000,000  
   private String seller; must be at least 2 non-blank characters long

*This class must have the following methods*

* 4 set methods,  one per instance variable. *15 points each method = 60 points*. These methods should validate the parameters and must call the HouseException if the parameters sent are invalid. The method names should be as specified below. Parameter names are up to you but the methods should take the parameter data type as indicated below. Follow the Java coding standards for set/mutator methods.
  + setMls(int mls)
  + setBedrooms(int bedrooms)
  + setPrice(double price)
  + setSeller(String seller)
* 4 get methods,*10 points each = 40 points.*Each one of these methods as you should know returns the value of an instance variable. You should know what the respective return data type for each one is.
  + getMls( )
  + getBedrooms( )
  + getPrice( )
  + getSeller( )
* 1 toString method, *20 points.* This method returns a String to be printed and takes no parameters, as you learned in ICS111. Each item/instance variable should be properly labeled. The price should be properly formatted with dollar sign and two decimals. Failure to do so will result in points deducted.
* 1 constructor method that validates the instance variables by calling the set methods. *20 points*. The call to set methods is to make sure that the parameters sent are valid while not duplicating code.
  + This constructor should take four instance variables as parameters, mls, bedrooms, price and seller.
  + Make sure that the parameters are ALWAYS in the following order (int mls, int bedrooms, double price,String seller). If parameters are not in this order, points will be deducted
* Overall program structure, naming conventions and style, following the Java coding standards *20 points*

**HouseException.java**(40 points total)

Make sure that you are not extending Throwable, you MUST extend Exception.

*instance variable*: String message

*This class must have the following methods. The constructor may be empty but must be there.*

* 1 constructor *10 points*
* 1 set method *10 points*
* 1 get method *10 points*
* Overall program structure, naming conventions and style, following the Java coding standards *10 points*

You *may*create a driver class to test your code. You do not need to submit the driver class with this assignment. Please watch the video where I explain how to test an object without a driver class [here: https://youtu.be/PmGr1muGTck](https://youtu.be/PmGr1muGTck)  The video is less than 2 minutes long.

**Please follow the Java Coding Standards for the class.**

<http://www2.hawaii.edu/~tp_200/ics111/material/codingStandards.html>

* You always must include your name as a comment at the top of your code.
* You must breifly comment all your methods.
* You must indent your code appropriately.
* You must use meaningful variable names.
* [just follow the coding standards!](http://www2.hawaii.edu/~tp_200/ics111/material/codingStandards.html)

**Your two classes (House andHouseException) working properly together** (20 points)

***If your code is incomplete or it does not compile you will get at least 60% deduction.***

Questions? Send an email to the instructor ASAP. Don't forget to include your full name and the class you are taking

**PositiveNumber.java**

public class PositiveNumber{

   private int num = 0;   // never public  
     
   public PositiveNumber (int num)throws Exception{  
      this.setNum(num);  
   }  
     
   public PositiveNumber ( ){  
      this.num = 1;  
   }    
     
   public void setNum(int num)throws Exception{  
      if(num >0){  
         this.num = num;  
      }  
      else{  
         MyException me = new MyException();  
         me.setMessage("The number is not positive");  
         throw (me);  
      }  
   }  
        
   public String toString( ){  
      String s = "the number is: " + this.num;  
      return s;  
   }  
     
   public int getNum( ){  
      return this.num;  
   }  
     
}

=====================================

### NiceOutput.java

import java.text.\*; //to access DecimalFormat  
public class NiceOutput{

   public static void main(String [ ] arg){  
         
      double value = 55555.6;  
      DecimalFormat formatter = new DecimalFormat("$###,000.00");  
      String output= formatter.format(value);  
      System.out.println("original number " + value);  
      System.out.println("nice output " + output);  
   }  
}

=====================================

### NiceOutput2.java

public class NiceOutput2{

   public static void main(String [ ] arg){  
         
      double value = 5.55555;  
      System.out.println("original number " + value);  
      System.out.print("using printf " );  
      System.out.printf("%.2f", value);  
       
   }  
}

=====================================

### MyException.java

public class MyException extends Exception{  
     //one instance variable  
   private String message ="he he it crashed!!!";  
     
   public void setMessage(String message ){  
      this.message = message;    
   }  
     
   public String getMessage( ){  
      return this.message;  
   }  
   
}

=====================================