Java **Assignment 5** **Due: see Blackboard**

Topic: Create a **Loan** class and use exceptions.

- Create and fully test a class named **Loan** which will have the functionality to produce various term payments on a typical loan. UML class diagram as follows:

Loan
- loanAmount: double - interestRate: double - durationYears: int
+ Loan() + Loan(loanAmount : double, interestRate : double, durationYears : int) + monthlyPayment() : double + monthlyTotalInterest() : double + biweeklyPayment() : double + biweeklyTotalInterest() : double + weeklyPayment() : double + weeklyTotalInterest() : double <all getters and setters>

Notes:

1. Note that the **interestRate** data field is to represent the annual interest rate as a number versus decimal form. For example, 7.25 % interest would be 7.25 and not 0.0725.

2. The default constructor will initialize all data fields to zeros.

3. The **three payment methods** will each return the appropriate payment for the loan provided that legal data is provided in the data fields. One illegal value for a data field would be if the **durationYears** is zero, in which case an exception will be thrown. Another illegal value for a data field would be if any of the three data fields were negative, in which case an exception will be thrown. (details below in paragraph 5)

4. The formula to be used to calculate a payment is as follows:

   Formula for a periodic payment is

   payment = **principal** * **rate** / (1 - (1 + **rate**) ^ (-**n**))

   where **principal** is the amount of the loan, rate is the **periodic interest rate** in decimal form, and n is the **periodic duration** of the loan. For a monthly payment; rate must be monthly interest rate and n will be the number of months of the loan.

5. The **three TotalInterest** methods will compute and return the total amount of interest that will be paid if the loan were to be paid off using that periodic payment method.

6. You will need to create two exceptions classes: **InvalidDurationException** and **NegativeLoanFieldException**, which will both inherit from the **Exception** class. The first exception will be thrown by any of the three payment methods if the **durationYears** is zero. The message that will be passed back with this exception will be: "***Duration of the loan cannot be***

zero." The second exception will be thrown by any of the three payment methods if any of the data fields is negative. The message that will be passed back with this exception will be: "***Loan fields cannot be negative.***"

7. In addition to the **Loan** class, you can choose to write a **TestYourLoanClass** class in order to fully test the functionality of the Loan class. However, a **JUnit** test class named **TestLoan** has been provided which fully tests the **Loan** class to ensure it works. If you execute the **TestLoan** JUnit test class (Ctrl+F6) and get no errors, you can be assured that your **Loan** class works properly.
8. Add appropriate **Javadoc** comments for the **Loan** class ensuring that you set the author as your name. Research the following Javadoc tags: @author, @param, @return, and @throws
9. Use the **javadoc** tool (Run → Generate Javadoc) to produce the HTML documentation for the **Loan** class.

Submit:

1. In Blackboard submit the following files: Loan.java, InvalidDurationException.java, NegativeLoanFieldException.java, TestYourLoanClass.java (or JUnit LoanTest.java) and a memo outlining your experiences with this assignment.

2. The following must be submitted in printed form: The output of your test class or a screen shot showing successful run of the JUnit test **and** the completed web page produced by **javadoc** for your **Loan** class.

Ensure you ask your instructor if you have any questions regarding this assignment!