# CS 46A Fall 2022

## Homework 06

### Requirements

1. You must name your classes exactly as specified. Otherwise Codecheck will not be able to process your submission and you will get no credit.
2. When you are finished with your code, submit it to Codecheck one final time then download the .signed.zip file.
3. You must upload all three signed.zip files together to Canvas and you should double check the files in Canvas to make sure all three zip files are uploaded.
4. Do not open the downloaded zip files. The files are digitally signed, and the grader program will check that they have not been opened.
5. Due time: 10 pm, Saturday, Oct 15. Submissions before the due time are not late.

   You will lose five points if your submission is marked Late in Canvas.

6. Grace time: 10 am, Sunday, Oct 16. Submissions before the grace time will not be rejected.

   You will receive no points if your submission is rejected by Canvas.

**Remember to follow our Programming Style Requirements!**

### Problem 6A

A life insurance company needs an application to automate the calculation of yearly life insurance premiums for individuals based on their gender and age and whether they live in a high crime area. If they live in a high crime area, there is a surcharge of 10%. The company will not insure a person 80 years old or older.

Here is a table of premiums based on age and gender of the person.

| | **Age < 21** | **Age >= 21 and Age < 60** | **Age >= 60 and Age < 80** |
|---|---|---|---|
| **Male** | $20 per year of life | $50 per year of life | $110 per year of life |
| **Female** | $18.50 per year of life | $45 per year of life | $100 per year of life |

For example: The premium for a male of 30 years old would be 30 * 50 if he lives in a low crime area and it would be 30 * 50 + (30 * 50) *0.1 if he lives in a high crime area.

Create a BlueJ project with three classes: **`Person`**, **`LifeInsurancePolicy,`** and **`LifeInsurancePolicyTester`**. Then copy class **`Person`** and **`LifeInsurancePolicyTester`** from Codecheck.

You need to look at the documentation of the **`Person`** class to make sure you can use the class in the **`LifeInsurancePolicy`** class correctly. You can look at either the Javadoc in the source code or the documentation for the **`Person`** class.

You need to write the entire class **`LifeInsurancePolicy`**. The class has two instance variables and one constructor with two parameters, an object of class **`Person`** and a boolean value, to initialize the instance variables. The boolean value indicates if the person lives in a high crime area or not.

Do not have instance variables for age and gender in the **`LifeInsurancePolicy`** class.

Implement the following methods for class **`LifeInsurancePolicy`**.

- **`public int policyHolderAge()`** Gets the age of the insured person. The age is stored in the Person object.
- **`public String policyHolderGender()`** Gets the gender of the insured person. The gender is stored in the Person object.
- **`public void incrementAge()`** Increments the age of the insured person. The Person class provides a method to do it.
- **`public double premium()`** Gets the yearly premium for the insured person or -1 if the age is greater than or equal to 80.

Requirements for method **`premium()`**:

1. Use nested if statements
2. Calculate the surcharge only in one place
3. Avoid calling any method more than once

Codecheck link for 6A

## Problem 6B

Create a BlueJ project with two classes, **`LoopyFun`** and **`LoopyFunTester`**, then copy class **`LoopyFunTester`** from Codecheck.

You will write the entire class **`LoopyFun`**. The class has one integer instance variable to store a positive integer. The class has one constructor with a parameter to initialize the instance variable. However, the parameter may not be positive, and the constructor should set the instance variable

to 1 if the parameter is 0 and to the absolute value of the parameter otherwise. You could use method Math.abs().

The class has three public instance methods:

- **`public int getNum()`** Gets the stored integer.
- **`public int factorial()`** Calculates and returns the factorial of the stored integer. For a positive integer num, the factorial is calculated as 1 * 2 *… * num. For example, for num = 5, the return value is 120 (1 * 2 * 3 * 4 * 5).
- **`public int sumOfPowersOfTwo()`** Calculates and returns the sum of powers of two. The power ranges from 0 (inclusive) to the stored integer (exclusive). For example, if the stored integer is 5, then the return value is 31 and is calculated as the following sum:

  $2^0 + 2^1 + 2^2\ 2^3 + 2^4 = 1 + 2 + 4 + 8 + 16 = 31$

You must use one loop in each of the methods **`factorial()`** and **`sumOfPowersOfTwo()`**.

Codecheck link for 6B

## Problem 6C

Create a BlueJ project with two classes, **`BookTitle`** and **`BookTitleTester.`** Then copy the code for **`BookTitleTester`** from Codecheck.

You need to write the entire class **`BookTitle`**. The class has one instance variable of String and one constructor with one parameter to initialize the instance variable. The parameter could have multiple words with some spaces, digits, and punctuation, but you can assume the string is well-behaved as described below:

1. It has one or more words separated by single spaces
2. It may have some spaces at the start and the end
3. Each word starts with a letter
4. For a word with two or more characters, one of the last two characters is a letter

The class has two public methods:

- **`public String firstLetters()`** Returns a string consisting of the first character of every word in the title.

- `public String lastLetters()` Returns a string consisting of the last letter of every word in the title. The trickiest part here is that if the last character of a word is not a letter as in “Absalom, Absalom!” you need to get the character before that.

You can also have the following private helper method:

- `private char lastLetter(String str)` Returns the last letter of a string, assuming one of the last two characters of the string is a letter.

You can call method `Character.isLetter(ch)` to decide if a character ch is a letter or not.

You must use one loop in each of methods `firstLetters()` and `lastLetters()`.

Codecheck link for 6C