Assignment 2

Reminder: Your code is to be designed and written by only you and not to be shared with anyone else. See the Course Outline for details explaining the policies on Academic Integrity. Submissions that violate the Academic Integrity policy will be forwarded directly to the Computer Science Academic Integrity Committee.
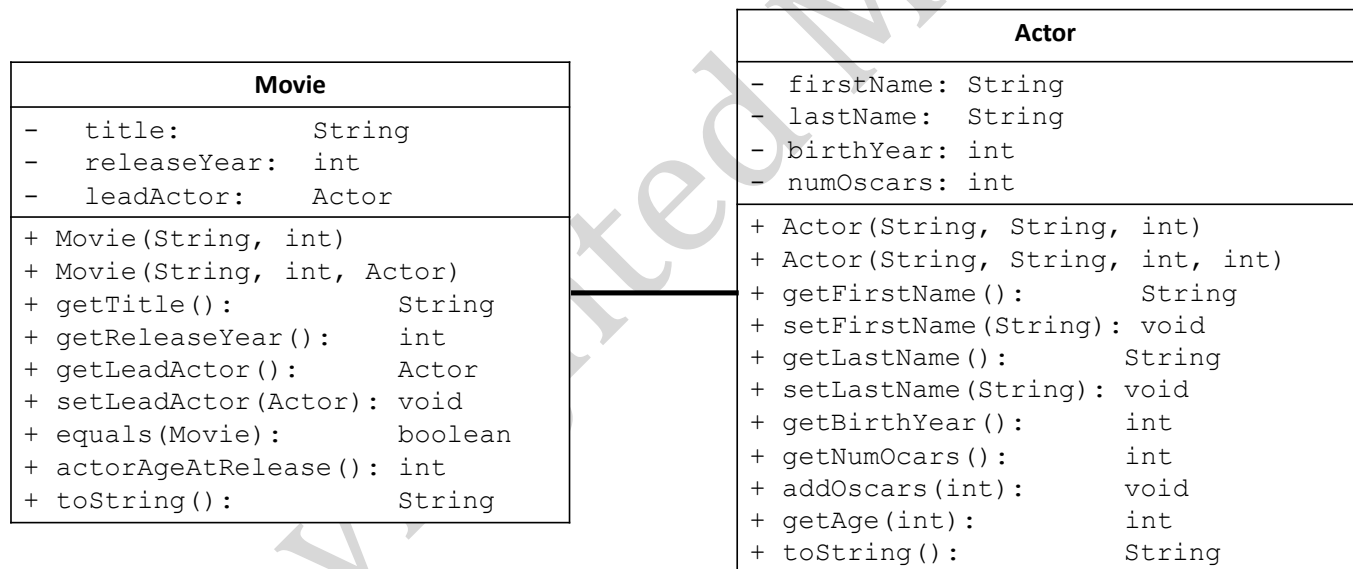
All materials provided to you for this work are copyrighted, these and all solutions you create for this work cannot be shared in any form (digital, printed or otherwise). Any violations of this will be investigated and reported to Academic Integrity.

Objectives
- Introduction to Objects
- More testing

Introduction
This assignment requires you to implement and test the two classes described in the following UML diagram. The line connecting the classes illustrates the relationship between `Movie` and `Actor`, that is, the `Movie` class **has a** field (`leadActor`) that is of type `Actor`.

Actor

- firstName: String
- lastName: String
- birthYear: int
- numOscars: int

+ Actor(String, String, int)
+ Actor(String, String, int, int)
+ getFirstName():   String
+ setFirstName(String): void
+ getLastName():   String
+ setLastName(String): void
+ getBirthYear():   int
+ getNumOcars():   int
+ addOscars(int):   void
+ getAge(int):   int
+ toString():   String

Movie

-   title: String
-   releaseYear: int
-   leadActor: Actor

+ Movie(String, int)
+ Movie(String, int, Actor)
+ getTitle():   String
+ getReleaseYear():   int
+ getLeadActor():   Actor
+ setLeadActor(Actor): void
+ equals(Movie):   boolean
+ actorAgeAtRelease(): int
+ toString():   String

NOTE: Given the `Movie` class has a field that is of type `Actor`, you should implement the `Actor` class first.

In this assignment you are provided with a tester. The auto grading of your assignment will include tests with different values.

Getting Started

1. Download A2Tester.java Actor.java and Movie.java to the same directory.
2. Read A2Tester.java and Actor.java carefully.
   a. Compile and run the A2Tester program from the directory you downloaded the files to.
      To compile: javac A2Tester.java
      To run: java A2Tester
3. Implement each method in Actor.java, by repeating the following steps:
   a. Uncomment a test in the corresponding method in A2Tester.java
   b. Add stubs for the methods in Actor.java
      NOTE: a stub is an empty method with only the signature and return value if there is one
   c. Compile and run to ensure your tests and stubs are correct
   d. Implement each method by completing the stub following the documentation
   e. Compile and run (repeat steps d and e until all of your tests on the Actor class pass)
   f. Repeat steps a-f for Movie.java

Submission and Grading

Submit the file named **Movie.java** and **Actor.java** with the completed methods through the assignment link in BrightSpace.

- You **must** name the methods in Movie.java and Actor.java as specified in the documentation (shown in the UML above) and as used in A2Tester.java or you will receive a **zero grade** as the tester will not compile.
- If you chose not to complete some of the methods required, you **must at least provide a stub for the incomplete method** in order for our tester to compile.
- If you submit files that do not compile with our tester (ie. an incorrect filename, missing method, etc) you will receive a **zero grade** for the assignment.
- Your code must **not** be written to specifically pass the test cases in the testers, instead, it must work on other inputs. We may change the input values when we run the tests and we will inspect your code for hard-coded solutions.
- **ALL late** and **incorrect** submissions will be given a **ZERO** grade.