ENSF 409
Winter Semester 2021
Assignment 05

Assignment Instructions

Complete and submit exercise 12.4 (Lesson 12). This assignment is due on February 22, 2021 by 5:59 PM Mountain Time.

General Instructions

Academic Integrity and Collaboration

This is an individual assignment. Your submission must be your own original work. You may not work with others to complete the assignment, although you may collaborate on exercises which are not submitted. It is not academic misconduct to ask for help on the discussion boards, but you may not copy code or complete answers from your peers.

Submission

You must submit your assignment in the specified format. Due to the number of students in the course, we are using automated grading. If your submission does not have the correct folder structure and file names, it will not be marked. This is a strict requirement.

File and directory names are case-sensitive.

You must submit your assignment to the appropriate D2L dropbox folder. You may submit multiple times before the assignment deadline, but only the last submission will be graded. Previous uploads are discarded by D2L. Therefore, each upload must be a complete submission. Do not submit multiple files across separate submissions.

The assignment must be submitted as a single zip folder named with your student ID. Furthermore, when the file is unzipped, the contents must be in a folder with the same name. You may wish to verify that your zip file has been correctly generated (includes the student ID directory, contains all files). You can do this by copying the zip file into another folder, unzip it, and examine the structure.

Within the folder with your ID number, you must create a subdirectory for each exercise within the assignment. Use lowercase and employ a _ to separate sub exercise numbers. For example, Exercise 1.3 should be in a folder `exercise1_3`.

Below is an example for submitting Assignment 1, assuming a student ID of *1234765*.

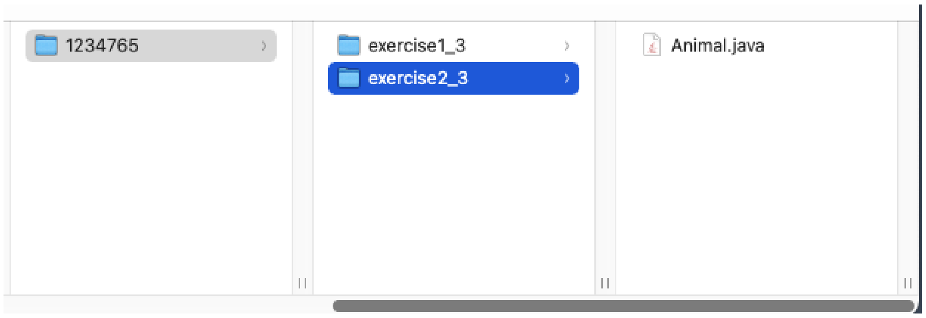
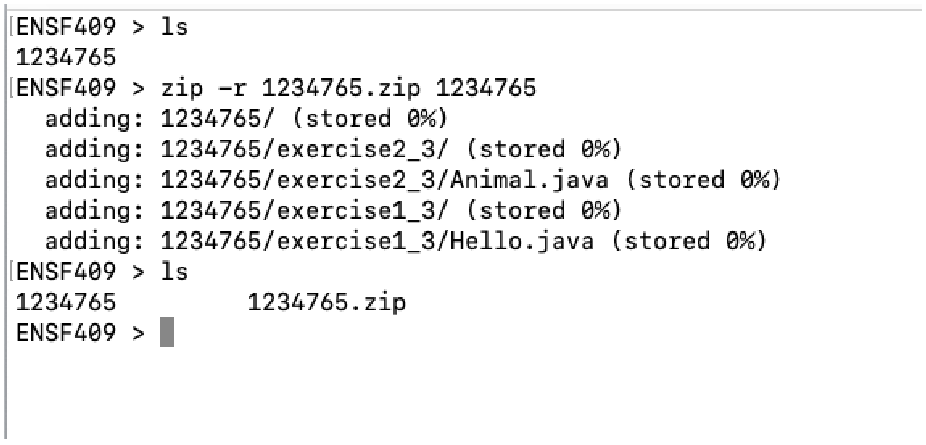
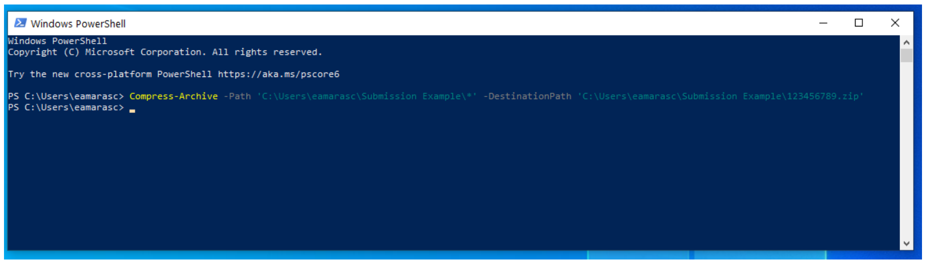


Figure 1: The folder structure for Assignment 1



```
ENSF409 > ls
1234765
ENSF409 > zip -r 1234765.zip 1234765
  adding: 1234765/ (stored 0%)
  adding: 1234765/exercise2_3/ (stored 0%)
  adding: 1234765/exercise2_3/Animal.java (stored 0%)
  adding: 1234765/exercise1_3/ (stored 0%)
  adding: 1234765/exercise1_3/Hello.java (stored 0%)
ENSF409 > ls
1234765         1234765.zip
ENSF409 >
```

Figure 2: Creating the zip file in Mac/Linux



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\eamarasc> Compress-Archive -Path 'C:\Users\eamarasc\Submission Example\*' -DestinationPath 'C:\Users\eamarasc\Submission Example\123456789.zip'
PS C:\Users\eamarasc>
```

Figure 3: Creating the zip file in Windows

Evaluation

Code which does not compile will not receive any points. Your code must compile and execute correctly to receive full marks. Assignments are graded using OpenJDK 11. While we do not anticipate that any of the assignments in this course will execute differently in any version of Java from 8 on, you should use this version for optimal results.

When style conventions are introduced in lessons, all subsequent submissions should adhere to the conventions.

Deadline

All homework assignments are due at 17:59 (5:59 PM) Mountain Time. You are responsible for the conversion between Mountain Time and your local time zone, if applicable. Be aware that the switch from Mountain Standard Time to Mountain Daylight Time occurs during the term.

It is recommended that you do not leave your submission until the last minute in case of technical issues. Once the dropbox folder closes, you will not be able to submit your assignment. Late submissions will not be accepted.

If there is a technical problem and you are unable to submit via the dropbox, do not email your assignment. The University of Calgary email system will remove zip files. Instead, you may email a link to download your .zip folder from OneDrive. The email must be received by Dr. Barcomb and Dr. Marasco before the assignment deadline and the OneDrive folder should not show any changes to the .zip file after the deadline has passed.

ENSF 409

Exercises - Lesson 12

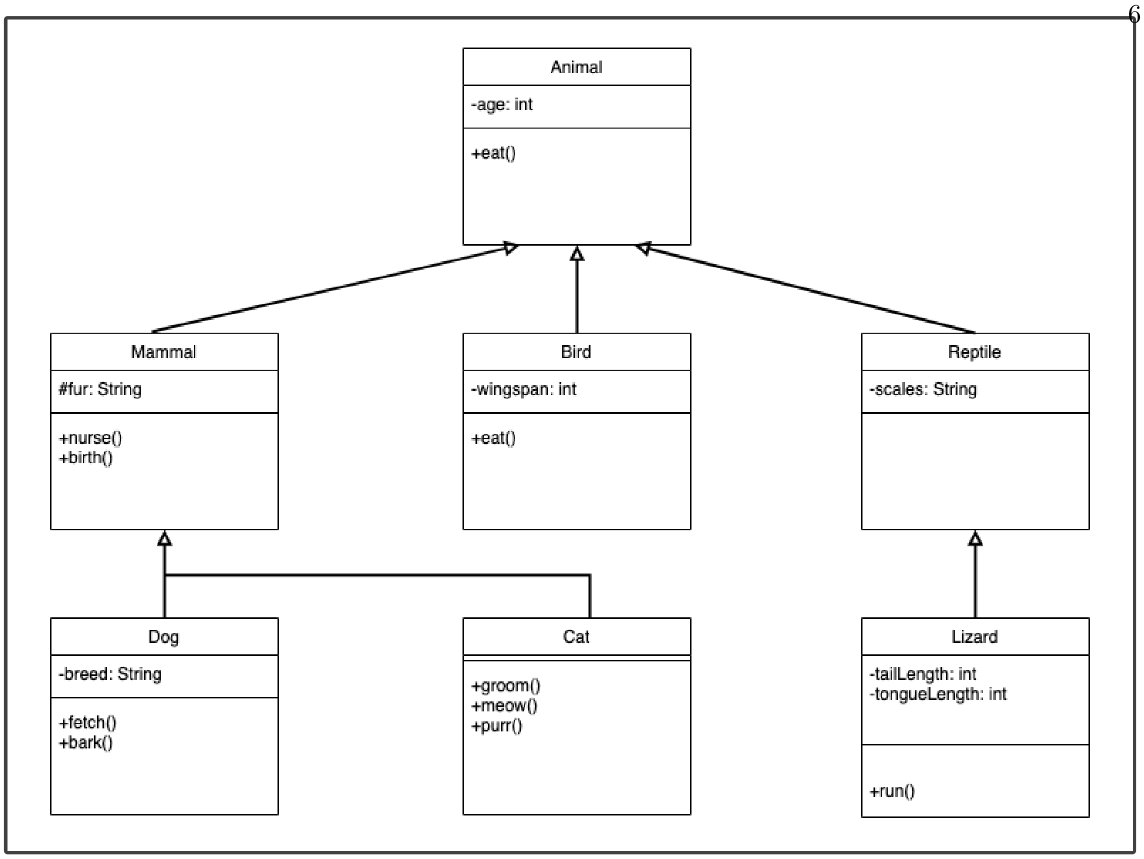
The following exercises are described in the video for Lesson 12.

Exercise 12.1

1. Create a public method `getYears()` in `MyExample.java` which returns this.age.
2. Change calls to `getAge()` to `getYears()` in the Reptile class.
3. Try to compile the code.
4. Change the variable age in the Animal class to protected.
5. Try to compile the code.

Exercise 12.2

1. Implement the class Lizard in `MyExample.java`.
2. Demonstrate retrieving `tailLength` and `tongueLength`. Include a call to `run()`.



Exercise 12.3

1. Create a UML class diagram representing an inheritance hierarchy between the different animals named. This diagram will contain methods but not variables.
2. Assume that there is no overriding, so in each class include only methods which aren't found in an ancestor class.
3. Add classes as needed to reduce the number of times methods are repeated. **Aim for the fewest possible methods.**
4. Do not use multiple inheritance.

Exercise 12.4

1. Write Java code in a file named MyBook.java which implements the UML diagram.
2. Getter and setter methods are not shown in the diagram but must be included, with standard names and functionality.
3. All non-getter methods which return String should return a String in the format format: *Method <methodname> called from <classname>*

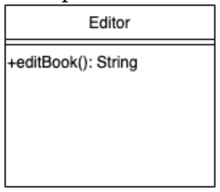
Tip: You'll need to create a public class matching the file name.

Tip: You should write only the constructors which are shown in the diagram.

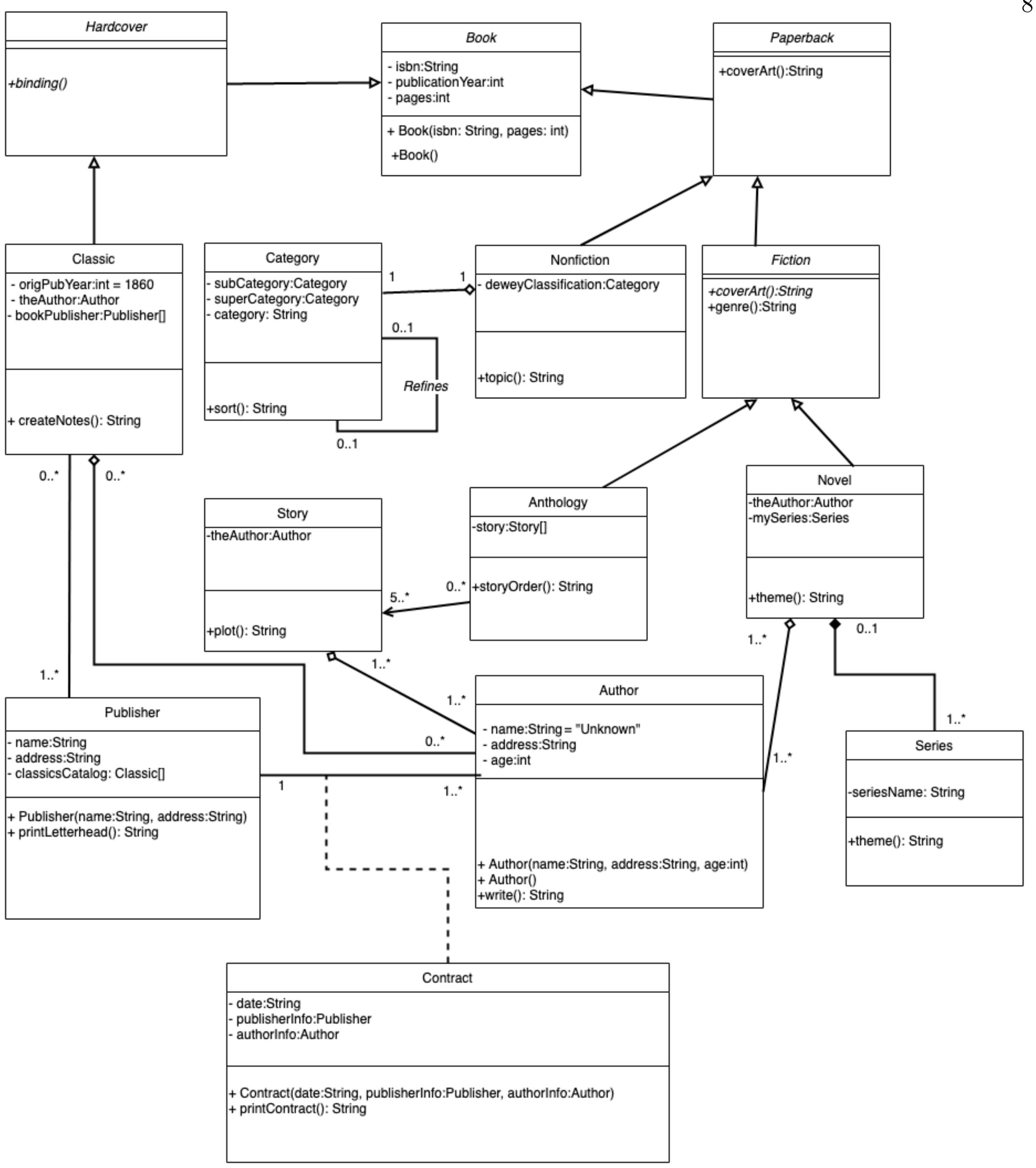
Important: Do not include extra spaces or punctuation in the returned Strings. Use the same capitalization shown here. The method name and class name are written as in the code, and sentence begins with a capital letter.

For this exercise, you can treat the upper boundary of * as 10.

Example:



```
class Editor {
  public String editBook() {
    return "Method editBook called from Editor";
  }
}
```


Hardcover
+binding()
Book
- isbn:String
- publicationYear:int
- pages:int
+ Book(isbn: String, pages: int)
+Book()
Paperback
+coverArt():String
Classic
- origPubYear:int = 1860
- theAuthor:Author
- bookPublisher:Publisher[]
+ createNotes(): String
Category
- subCategory:Category
- superCategory:Category
- category: String
+sort(): String
1
1
0..1
Refines
0..1
Nonfiction
- deweyClassification:Category
+topic(): String
Fiction
+coverArt():String
+genre():String
0..*
0..*
Story
-theAuthor:Author
+plot(): String
5..*
0..*
Anthology
-story:Story[]
+storyOrder(): String
Novel
-theAuthor:Author
-mySeries:Series
+theme(): String
1..*
0..1
1..*
1..*
1..*
Publisher
- name:String
- address:String
- classicsCatalog: Classic[]
+ Publisher(name:String, address:String)
+ printLetterhead(): String
0..*
1
1..*
Author
- name:String = "Unknown"
- address:String
- age:int
+ Author(name:String, address:String, age:int)
+ Author()
+write(): String
1..*
1..*
Series
-seriesName: String
+theme(): String
Contract
- date:String
- publisherInfo:Publisher
- authorInfo:Author
+ Contract(date:String, publisherInfo:Publisher, authorInfo:Author)
+ printContract(): String