**[CSE 007]** Classwork: Classes & Objects Assigned: 18 April 2022 **Objective:** In this activity, you will practice creating a class that defines a TVShow object type and creating various TVShow instances to practice accessing object members.

**Submission:** *TVShow.java, TVTest.java*

**The first pillar of object-oriented programming is data encapsulation**. Since we aim to write code that is robust, we are keeping our object definition separate from implementation. As such, you will begin by creating one file *TVShow.java,* that will work with *TVTest.java* (provided in Wk 12 resources; which you will modify).

* *TVShow.java* will define public class TVShow{}, but it will **not** include a main method. We will use a separate class/program to test our object classes, thus keeping the implementation details (the class that defines the object, TVShow) separate from the use of a class (the main method or method that includes creating and accessing objects).
* *TVTest.java* defines public class TVTest{}, which includes a main method (making this the main/executable program). Within this main method, objects of type TVShow will be created and used.

Using your basic understanding of classes & objects, implement TVShow to meet the following requirements:

* Declare the following data fields for your TVShow (all should be private):
  + String name;
  + String stream;
  + String genre;
  + double avgLength;
  + int episodes=0;
    - When should episodes increment?
  + **static** int shows=0;
    - Why is shows static? When should it increment?
* Create two constructors:
  + A no-arg constructor: public TVShow(){ … } that has no parameters, and initializes all of the data field values
    - Initialize each String to be “n/a” and avgLength to be 0
    - You should also increment shows here (why?)
  + A four-argument constructor:

public TVShow(String name, String stream, String genre, double avgLength)

**Ex:** TVShow(“Friends”,”HBO”,”Comedy”, 22.5)

* + That takes in 4 parameters, and initializes the data field values using the arguments passed in using this command if necessary:

this.name = name; this.stream=stream;

…

* + - You should also increment shows here (why?)
* Create getter (accessor) methods to return the private data field values
  + Ex: public int getEpisodes(){ return episodes;}
* Create setter (mutator) methods to set the value of each private data field
  + Ex: public void setEpisodes(int episodes){ this.episodes = episodes;}
* Define a method called airEpisodes(){} that increments the number of episodes aired so far of the calling TVShow object.
  + What is the likely return type of airEpisodes()?
  + What parameter(s) (if any) are expected?
* Overload airEpisodes() to accept an integer value; rather than incrementing episodes by 1 each time, it will be increased by the value passed into the method.
  + For example: tvshow1.airEpisodes(5) would add 5 to the current value of episodes for the tvshow1 object.
* Define getTotalAirtime(){} that will return the total amount of airtime used by a given TVShow object. (Hint: episodes \* avgLength)
  + What is the likely return type of getTotalAirtime()?
  + What parameter(s) (if any) are expected?

Download and save the main program (*TVTest.java*) from the Week 12 Resources.

Within the main method, you will find code that declares and creates an array of 4 TVShow objects -- each object reference created using the 4-argument constructors.

Begin by:

* Using a for-loop to iterate through the TVShow array to call airEpisodes(x) in order to increase the number of episodes for the TVShow objects, passing a random number into the method.
* Creating a separate print method that uses various calls to

System.out.println() to display the following:

* + The number of TVShow objects created (accessed via TVShow.shows)
  + The number of episodes and total air time for each object created

*Note:* Make sure your two classes are in the same folder at compile time. Also, remember that even though you need to compile both classes, only *TVTest.java* will be run.

*\*Challenges*:

* Declare all of your data fields (including the static one(s) as private). Implement getter and setter methods as necessary to adjust for these changes.
* Define a print() method within TVShow.java to display important information related to a TVShow object (and modify your TVTest accordingly).