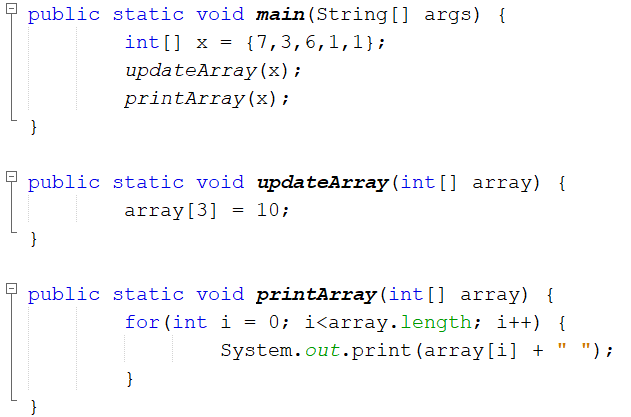
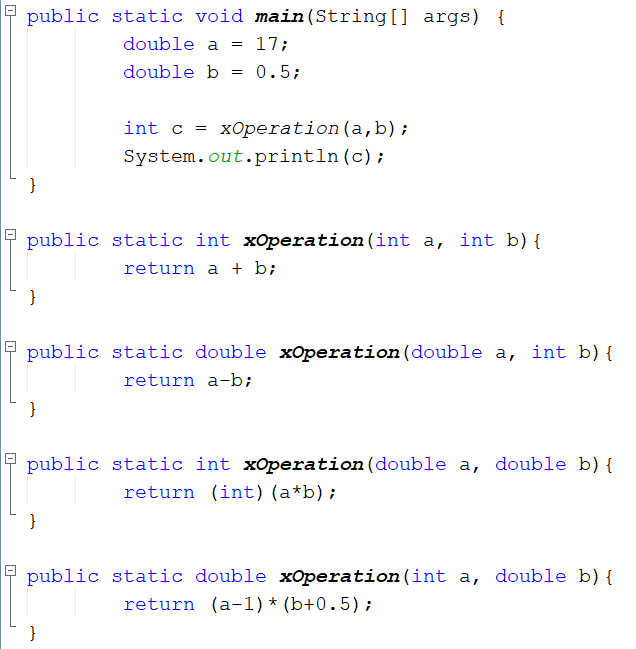
Instituto Tecnológico y de Estudios Superiores de Monterrey  
Computer Science II – 2nd Partial Lab

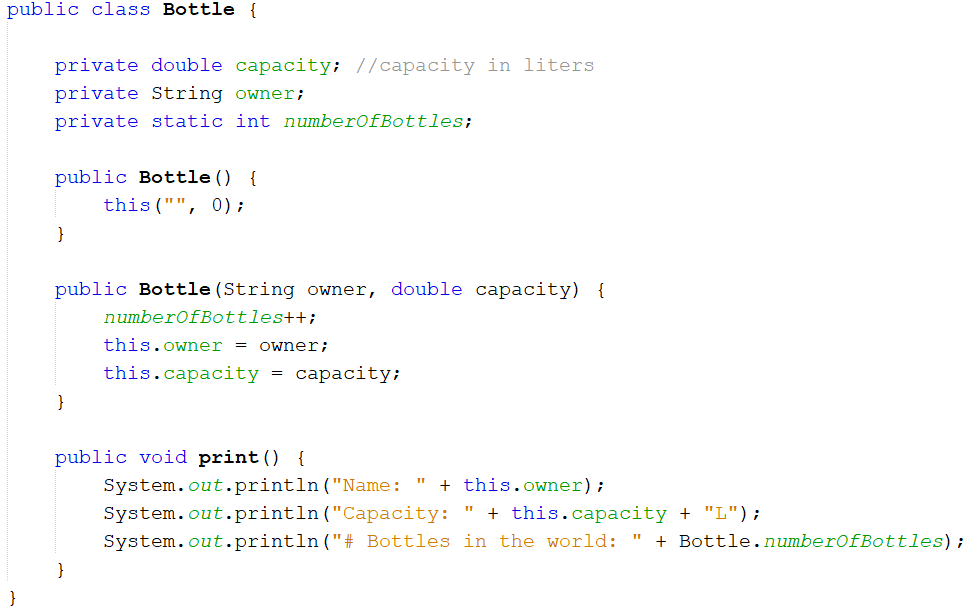
**Section 1. Select the right option.**

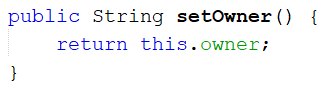
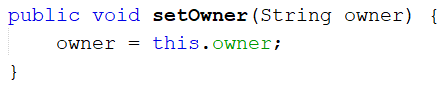
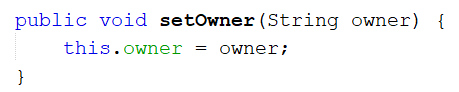
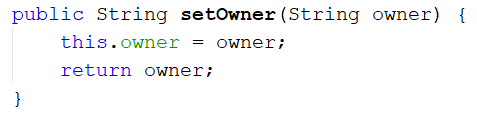


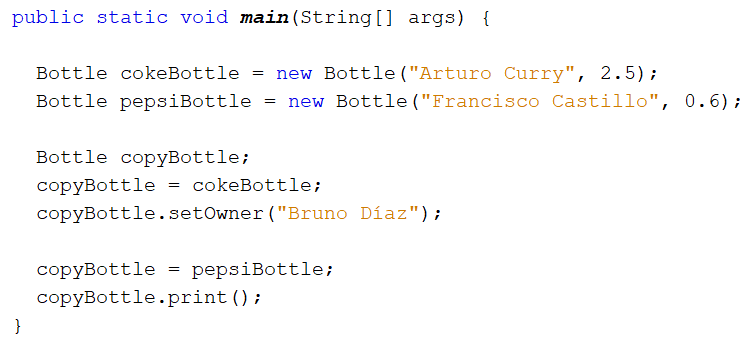
* 1. 7 3 6 1 1
  2. 7 3 10 1 1
  3. 7 3 6 10 1
  4. 7 3 10 1



1. 17.0
2. 8
3. 16.0
4. 16
5. 17
6. 8.0

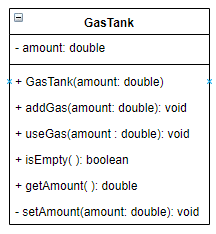


1. **Choose the best setter for the owner variable:**
   1. 
   2. 
   3. 
   4. 
2. **What would be the result of running the code below:**



1. Name: Francisco Castillo  
   Capacity: 0.6L  
   # Bottles in the world: 2
2. Name: Arturo Curry  
   Capacity: 2.5L  
   # Bottles in the world: 1
3. Name: Francisco Castillo  
   Capacity: 0.6L  
   # Bottles in the world: 1
4. Name: Bruno Díaz  
   Capacity: 0.6L  
   # Bottles in the world: 1

**Section 2:** Design a class called **GasTank** that represents a tank of gasoline:



* **Constructor:** Method to initialize the instance variable **amount**. Use the method **setAmount**.
* **addGas**:Increases the amount of gas by the amount received as a parameter. Validate that only positive values are processed.
* **useGas:** Reduces the amount of gas in the tank by the amount received as a parameter. Validate that only positive values are processed.
* **isEmpty:** Should return **true** when the amount of gas in the tank is smaller than 0.1. Otherwise, return **false**.
* **getAmount:** Getter for the **amount** variable.
* **setAmount:** Private method (should only be used by the Constructor) to set the value of **amount.**

**Section 3.**

Cineplus, a new movie theater complex, has decided to open its first movie theater in Monterrey. Its unique concept is to offer tickets with dynamic pricing depending on how many seats are available at purchase time.

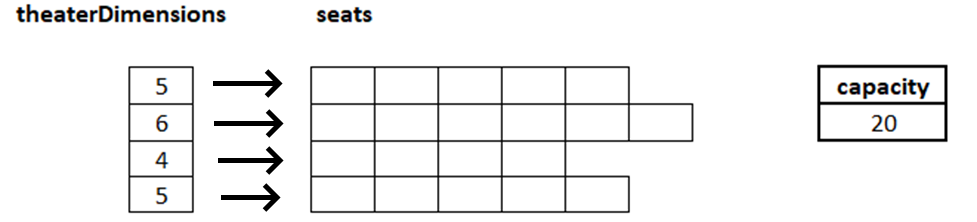
Code a class CinemaShow that allows the user to model a show at Cineplus. The class should have the attributes defined next. Use the best practices of object-oriented programming. Consider the proper visibility (public, private) and if every method or variable should be instance or static.

* Variable movieName to represent the movie that is being screened.
* Variable movieDate to represent the date when the movie will be screened.
* Variable movieTimeto represent the time when the movie will be screened.
* Variable capacity to store the capacity of the movie theater (how many people the show can store).
* Variable soldTickets to keep track of how many tickets have been sold at any given point.
* Multidimensional array seatsto keep track of the tickets sold.

Additionally, include the following methods:

* Constructor method to initialize: movieName**,** movieDate,andmovieTime**.**
* Setter methods for movieName**,** movieDate,andmovieTime.
* Getter methods for capacityandsoldTickets**.**
* Method initializeSeatsthat received the theater dimensions as an array and initialize the seatsmultidimensional arrays. Additionally, once seats is initialized, capacityshould be updated.

The theaterDimensionsarray represents the number of seats for every row of the theater. For example:



* Method assignPrice(int soldTickets, int totalTickets) that calculates and returns the price for the next ticket sold using the following formula:

For example, the ticket #16 in a theater with a capacity of 20 seats would be sold:

* Method sellSeat(int row, int column)that receives two integer inputs: row and column and validates the seat is empty. If it isn’t, it should immediately return false. If it is empty, it should calculate the price for the ticket using the method assignPrice, then store it on the row/column combination received as input and proceed to update the soldTickets counter**.**