1. 1. List and describe the purpose of each class that will be needed to solve this problem
   1. You must separately identify (list) and describe the purpose of each class. One or two sentences per class should be sufficient. You do NOT need to include the CircleDrawer class, since this class was provided for you.

courseRegistration getters:

they are used to get the values that have been set for the object CourseRegistration.

courseRegistration setters:

they are used to set the values for the object CourseRegistration.

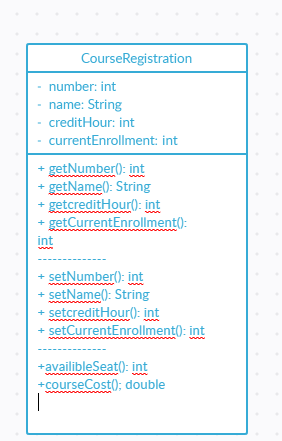
availibleSeat():

this is used to calculate the number of available seats based off the current enrollment.

courseCost():

this is used to calculate the cost of the course based off the credit hours entered.

1. 2. *Data Definition Class(es)* - Create a detailed UML Class Diagram, listing and explaining all class variables, accessors, mutators, special purpose methods, and constructors associated to each data definition class.



1. *Implementation Class* - Create a table that lists all methods that will be used to create the implementation class. For each method identified, provide the following:
   1. A 1-2 sentence describing the purpose of the method
   2. A list of the names, data types, and brief description (1-2 sentences) for each input variable into the method, if there are any
   3. The name and data type of the variable to be returned from the method, or void if nothing will be returned

Method: getNumber

Purpose: returns the number of students.

Input: none

Return: number: int

Method: getName

Purpose: returns the course name

Input: none

Return: name: String

Method: getcreditHour

Purpose:returns the course credit hours

Input: none

Return: creditHour: int

Method: getCurrentEnrollment

Purpose: returns the current course enrollment

Input: none

Return: currentEnrollment: int

Method: setNumber

Purpose: sets the number of students in the course

Input: number: int

Return: void

Method: setName

Purpose: sets the name of the course

Input: name: String

Return: void

Method: setcreditHour

Purpose: sets the number of credit hours for the course

Input: creditHour: int

Return: void

Method: setCurrentEnrollment

Purpose: sets the current enrollment of the course

Input: currentEnrollment: int

Return: void

Method: availibleSeat

Purpose: returns 30 minus currentEnrollment

Input: none

Return: availibleSeat: int

Method: courseCost

Purpose: returns creditHour times 670.00

Input: none

Return: courseCost: double