## Part I –

Information about a College object includes its name, its tuition, and the region in which it was located. You will write a College class whose state includes its name, tuition, and region. Please complete the following:

1. Declare and compile a public empty College Class. Don’t forget to append //end of class.
2. Declare the three instance variables to describe the state. Variables are declared as lower case words.
3. What accessor types should you use for the variables?
4. What type of variable is associated with each instance variable declaration?
5. Write a no parameter constructor (default constructor). Be mindful that constructors have no return type, are public, and are the same name as the class.
6. Write an overloaded constructor. Its parameter list should include the same types of variables, in order, as its state. Let’s name them myName, myRegion, and myTuition. What type is each parameter?
7. Initialize the state of the object (name, region, tuition) using its instance variables. Assign each parameter value to its corresponding state variable.
8. Compile before moving on.
9. Write the following accessor methods: getName, getRegion, and getTuition.
10. Write a mutator method updateTuition. It takes an integer parameter, newTuition that updates the tuition state of the object.
11. Write a toString method that returns that state of the object. Separate words by blank spaces.

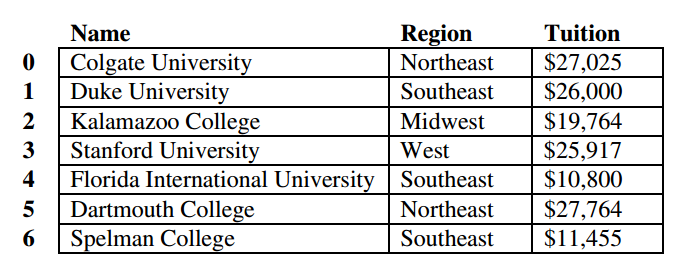
*Example of the empty class (see step 1).*

public class College  
{

}//end of class

## Part II –

The following chart shows an example of colleges that could appear in an object of type CollegeGroup.



1. Write the CollegeGroup class. It should have a private College array variable, colleges.
2. Write a CollegeGroup constructor that takes in a College array named myColleges as a parameter.
3. Inside the constructor, first construct the empty colleges array of the same length as myColleges.
4. Copy all College objects from myColleges into colleges via a for loop.
5. Don’t forget to compile!
6. Write a printColleges method to print out all College objects in colleges. What is its return type?
7. Write an algorithm to do step 8 before implementing the actual code. Use your own words to describe it.
8. Write the method updateTuition, which is described as follows:
   1. updateTuition will change the tuition of a College object in the CollegeGroup
   2. The parameters it passes are collegeName and newTuition
   3. What are the parameter types?
   4. What is its return type?
9. Write an algorithm to do step 10 before implementing the actual code. Use your own words to describe it.
10. Write the method getCollegeList will check each College in the array myColleges. The College object is added to a new array if it satisfies the following conditions: the College is in the region given, and the tuition is between low and high, inclusive.
    1. getCollegeList will return an array of College objects
    2. The parameters it passes are myRegion and two integers low and high
    3. What are the parameter types?
    4. This method will check each College and add it to a new array of type College[]
    5. The new array MUST be the same size as the number of colleges that satisfy the parameters
    6. Last the method returns the new array of College objects

## Part III –

Follow up questions –

1. Why do we use two separate for loops to traverse the array colleges in the method getCollegeList?
2. Why would returning an ArrayList be easier than an array?
3. When should you instantiate the new College[] array to the correct size?
4. It’s much easier to read and understand code that uses a separate counter to traverse and populate the new College[] array result from the method getCollegeList. How could you use the original counter to do this without starting over?