# Programming Assignment

**Scenario:**

You have been hired as an associate for a Real Estate Portfolio company specializing in renovations. Your first task is to create a program that calculates the break-even cost for each property under their portfolio. The Real Estate Portfolio currently has 5 properties under their portfolio. Each property has the following characteristics: an address, purchase price, and a renovation cost.

At minimum, each property must have an address. The purchase price of the property must be greater than zero and the renovation cost should be no greater than the purchase price. In order to calculate a property’s break-even cost, you must enter the property’s tax percentage for the area (for example, the tax percentage could be 5%). The tax percentage can only be as high as 33%. Use the below formula to calculate the break-even cost for each property.

 **Break-Even Cost Formula**Purchase Price + Renovation Cost + (Purchase Price \* (1 + tax percentage))

Your object-oriented application must use a data definition class to model each Property with an implementation class requesting each properties characteristics and the tax percentage. The data definition class should track the address, purchase price, and renovation cost. Accessors and mutators must be created for all instance variables.

With a completed data definition class, your implementation class should only define 1 property object, but be reused to instantiate each property. Your object-oriented application will gather the properties address, purchase price, renovation cost, and that property’s tax percentage. The application should then print out well-formatted report containing the property address, purchase price, renovation cost, and break-even cost. Then, the application should repeat this process until all 5 properties have been entered. Once all properties have been entered, print out a final well-formatted message stating the average purchase price, renovation cost, and break-prices for all the properties.

**Other Requirements:**

* Your solution must use object-oriented techniques (No points earned for a procedural solution).
* Your solution must be designed as a modular solution using methods other than main, with each method performing one task. The main method should contain only a very minimal amount of code, calling methods instead.
* Your solution must include appropriate constants, constructors, accessors, mutators, and special purpose methods with exception handling, as necessary in support of the problem.

**Hints:**

* You do not need to use arrays to solve this problem. Keep in mind once you have finished entering the information for one property, it is not necessary to keep the information.
* Think about what type of validations might be appropriate for this application. Make sure these are all handled.
* Think about any special purpose methods that might be needed in the data definition class. Special purpose methods are important when using values in instance variables to perform a specific process.

 **Programming Assignment : Solution Implementation**

Write a well-documented, efficient Java program that implements the solution design you identified. Include appropriate documentation as identified in the documentation expectations document.

To Blackboard, **submit ONLY ONE .zip file** containing all of the .java files part of your submission for your solution implementation. As you are naming your Java files, make sure they are indicative to the purpose of the file. **Do not include your name in the file name or use generic names, such as Assignment2.java**. Your .zip file should contain only the .java files in your solution. Be careful that you do not submit .class files instead of .java files.

**Warning!** You must submit **ONLY** **ONE** .zip file containing **ONLY** your .java files. Failure to follow this instruction precisely will result in a 10 point deduction of the assignment score. **No exceptions!
*Why is this important?* The goal is to teach you how to properly package your IT solutions into a “customer-friendly” format while paying attention to “customer” requirements provided to you.**

**Your program must compile using jGrasp**. Any final program that does not compile, for any reason, will receive an automatic zero. Other IDEs often place in additional code that you are unaware of, doing too much of the work for you. **You are strongly discouraged from using IDEs other than jGrasp.**

|  |
| --- |
| **Grading Criteria** |
| **Requirement** | **Points** |
| Implementation of object-oriented Java program, using efficient practices, such as the use of constants, good variable names, information hiding, no redundant code, etc. | 80 |
| Appropriate objective-style documentation | 10 |
| Appropriate intermediate comments | 10 |