# CST8110 Assignment #3 [5%]

**This assignment relates to the following Course Learning Requirements:**

CLR 1: Define, analyze, and document the logic of a solution to a given problem.

CLR 2: Implement the solution to a given problem by writing the appropriate code in a high-level language (Java).

CLR 4: Install and use the Java Development, Runtime Environment, and documentation libraries.

CLR 7: Create documentation and a Java solution for programming problems that adhere to the submission standard identified within the timeframe given in the problem description.

Objective of this Assignment:

The following is an exercise to help you understand what you have learned in this module. It will allow you to familiarize yourself with the module content as well as practice the skills required to develop software using the Java programming language. The objective of this assignment is to write a Java application to calculate the shipping costs of two packages using a cost formula and output a comparison of the two costs.

There are several ways to design the logic for this assignment, however if the logic receives the correct result, it is correct!

# Pre-Assignment Instructions:

1. To prepare you for this assignment, please complete the content associated to this module.
2. Complete all module reading and learning activities
3. Complete all reference video tutorials
4. Develop this program locally on your machine in a plain text editor or other IDE

**Assignment Tasks/Steps:**

1. Create a Java class named **Package** that contains the following:
2. Package should have three **private** instance variables of type **double** named **length, width, and height**.
3. Package should have one privateinstance variable of the type **Scanner** named **input**, initialized to **System.in**.
4. **No-args** (explicit default) public **constructor**, which initializes all three double instance variables to 1.0.
5. **Initial** (parameterized) publicconstructor, which defines three parameters of type double, named length, width, and height, which are used to initialize the instance variables of same name.
6. Public **copy** constructor, with a parameter of type Package, used to duplicate an existing Package object.
7. Three public void methods named **inputLength**,**inputWidth**, and **inputHeight**. Each method will prompt the user for the appropriate property, and input a double value using the Scannerobject input to initialize the instance variables
8. A **public void** method named **displayDimensions** which prints the dimensions as length X width X height (each value separated by a “ X “).
9. A public method of type double named **calcVolume** that calculates the volume and returns the result as a double value.
10. Create a class named **Shipment**

**These points describe the behaviour and the goals of this class, for the variables and methods of this class see the figure on the next page. You have to match these goals with the corresponding methods of this class.**

1. The program mustask for two separate packages to ship
2. The program mustcalculate the cost difference using the difference in volume:

i.e. The base price for a package with volume <=1 is $3, for every unit increase in volume, the cost increases by $1

e.g. 1: a parcel with volume 4, the cost is $3+ $1 +$1 +$1 = $6

e.g. 2: a parcel with volume 2.5, the cost is $3 + $1.5 = $4.5

1. Give the following (in order of priority):
2. If there is no difference, display the costs as the same
3. If the cost of one is less than twice the other, display that it is “slightly more than”
4. If the cost of one is less than three times the other, display that it is “twice”
5. If the cost of one is less than four times the other, display that it is “triple”
6. If the cost of one is less than five times the other, display that it is “quadruple”
7. Otherwise, display that as a calculated multiple
8. The program mustindicate the more costly package (if not same cost) and by how much
9. The program mustcalculate and display the appropriate message (including proper dimension and cost format)
10. Your code also does NOT (YET) need to worry if the user inputs an invalid value for the input (example: invalid length). We will implement this in a future assignment.
11. The output of your code mustmatch the samples.
12. You must change the title i.e. Welcome to John Doe’s shipping calculator!
13. Create a class called **Assignment3** contains only the main method and the following lines of code:

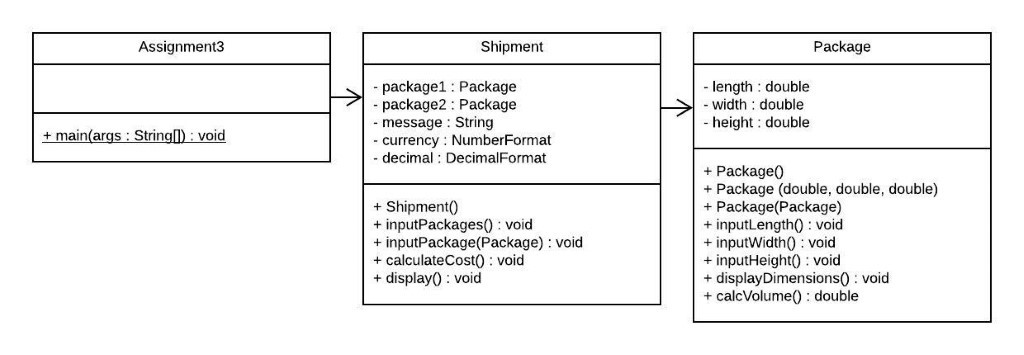
Shipment s = new Shipment();

s.inputPackages();

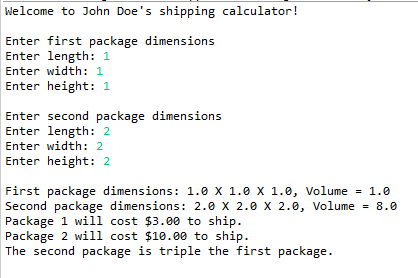
s.calculateCost();

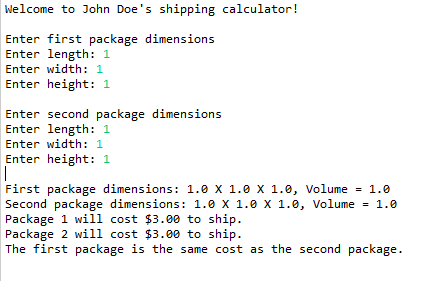
s.display();

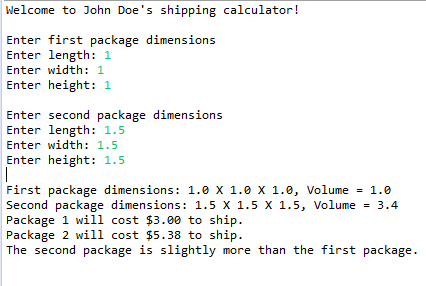
UML Class Diagram (**This diagram shows the contents (variables and methods) for each class**)



**Sample Outputs**







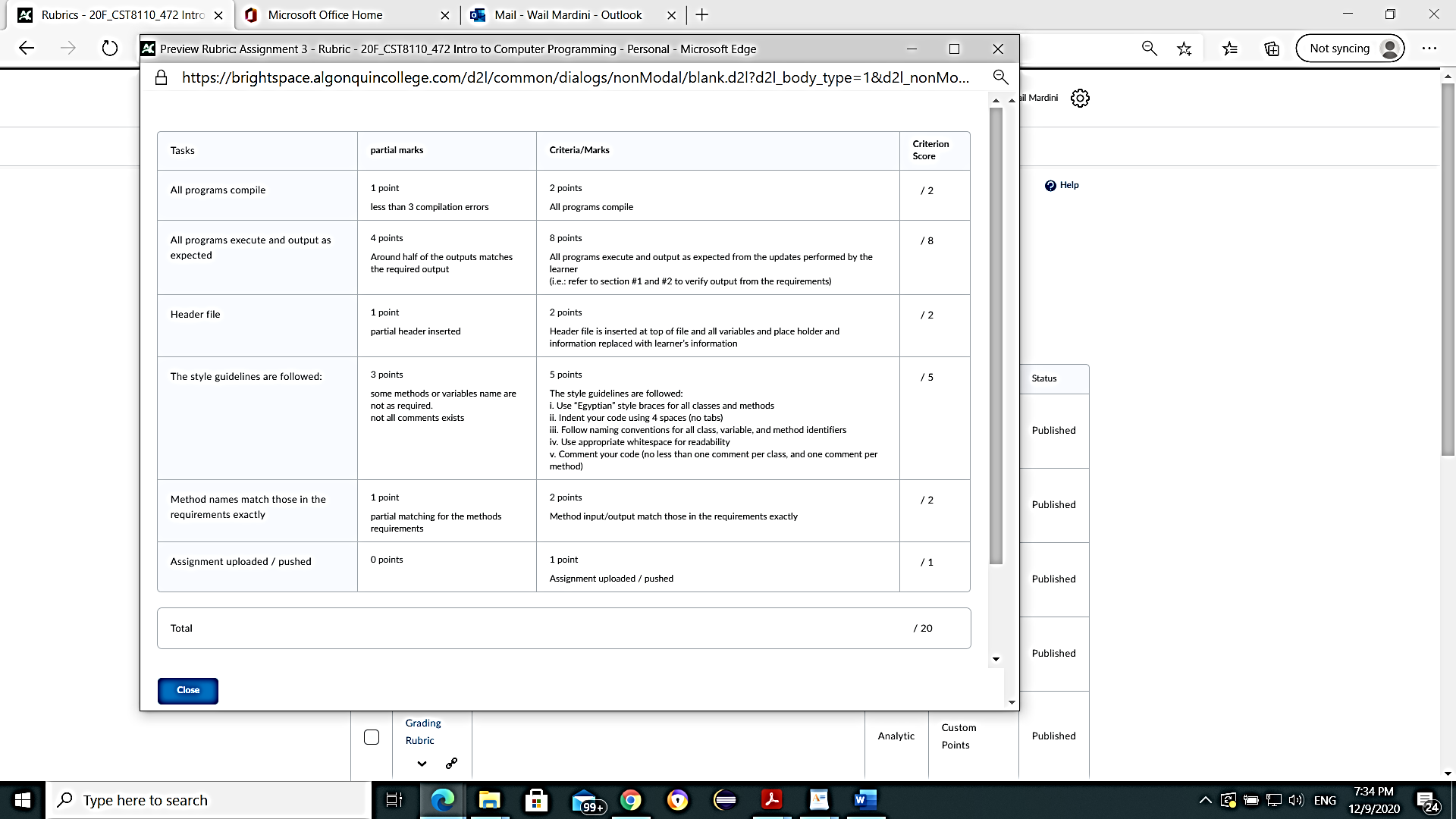
**Submit Your Assignment**

Your assignment is to be submitted on Brightspace as Java files. It should be submitted with the following guidelines:

1. Include the file header using the template provided in Assignment 1, in every java file submitted.
2. Follow expected style guidelines:
3. Use “Egyptian” style braces for all classes and methods.
4. Indent your code using 4 spaces (no tabs).
5. Follow naming conventions for all class, variable, and method identifiers.
6. Use appropriate whitespace for readability.
7. Comment your code (no less than one comment per class, and one per method).
8. Files must be named **Assignment3.java**, **Package.java** and **Shipment.java**.
9. Method names must match those in the requirements above, exactly.

Feel free to make any assumption in order to implement a working code. Make sure the code delivers expected successful output. Highlight any assumption you may make while submitting the assessment.

**Assignment Grading Rubric (5%)**



**20 points will be mapped to 5% of the total course mark**