CS 230 Module One Assignment Guidelines and Rubric

Overview

Software is more about design and organization than about the language and syntax rules of curly braces and semicolons. You should be able to look at a UML class diagram and picture the code that is built from it. Conversely, given a set of classes, you should be able to create a UML class diagram from them. There are even commercial tools that can do that translation back and forth for you: from diagram to code and back.

Now that you have downloaded the Java Development Kit (JDK) and the integrated development environment (IDE), Eclipse, to your desktop, you will use this as an opportunity to examine the source code files to create a UML class diagram.

Prompt

Getting Set Up

Please note: You should have completed these steps during the Technology Setup task.

1. If you have not already done so, download the zipped file folder that contains the source code files from *Hands-on Design Patterns With Java*: Chapter 2 (Bicycle.java, Driver.java, TwoWheeled.java, and Vehicle.java). Please note: The Driver.java class is the program driver.
2. Open your IDE, Eclipse, and create a new Java project.
3. Upload the zipped file folder with the Java code into the Eclipse project. A tutorial is available for uploading files to Eclipse: Uploading Files to Eclipse Desktop Version Tutorial PDF.

UML Class Diagram

4. Examine the source code and classes.
5. Develop a UML class diagram for the application using Lucidchart. A tutorial for creating a free Lucidchart account to design UML diagrams is available: Lucidchart Tutorial for Creating Class Diagrams PDF.
6. Your class diagram must include all classes and demonstrate the relevant object-oriented programming principles.
7. Briefly describe the object-oriented programming principles illustrated in the class diagram.

What to Submit

Submit a Microsoft Word document that includes an image of the UML class diagram you created and a brief write-up.

Module One Assignment Rubric

Criteria	Proficient (100%)	Needs Improvement (70%)	Not Evident (0%)	Value
UML Class Diagram	Creates a UML class diagram that depicts all properties of and relationship between multiple classes for a given application	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include missing properties or relationship between classes	Does not attempt criterion	50
OOP Principles	Describes the object-oriented programming principles illustrated in a class diagram	Shows progress toward proficiency, but with errors or omissions	Does not attempt criterion	40
Articulation of Response	Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose	Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability	Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas	10
			Total:	100%