CIS 1400: Programming Logic and Technique
Group Project

Group Project	#3 – Class Design
Team Number	
Team Members	_____ _____ _____ _____ _____
Flipgrid URL (*select 'Share' button to get private link*)	
Team Points	Class UML Diagram _____/ 10 pts. Class Pseudocode _____/ 15 pts. Video Presentation _____/ 5 pts. **Total _____/ 30 pts.**

Group Project Instructions

Each team will **collaboratively** design a class to solve an assigned problem. As part of the design and project grading, each team will create a short (< 5 minutes) screen recorded Flipgrid video presenting the following:

- Students in the group and their contributions
- Class UML Diagram
- Class Pseudocode

Grading criteria includes proper class design format (UML and pseudocode), and descriptive class, attribute, and method naming. Grading will be done through video viewing (***ensure design documents are viewable during video***); **only** submit this completed project assignment form through Blackboard by the Due Date/Time. Other uploaded files in Blackboard will incur a 20% grading penalty.

Some helpful resources to get you started:

- Using Blackboard Groups: https://help.blackboard.com/Learn/Student/Interact/Groups
- Flipgrid Getting Started for Students: https://help.flipgrid.com/hc/en-us/articles/360051542894
- Flipgrid Help Center: https://help.flipgrid.com/hc/en-us
- Link to Flipgrid Topic: https://flipgrid.com/6a991db4(use your @dupage.edu or @cod.edu email to login; *password is cis1400vcm01*)

CIS 1400: Programming Logic and Technique
Group Project

Assigned Team Problems:

Team Number	Problem
1	Design the class implementation for a **House** that includes the following **private** data members (*data type in parentheses*): • address (string) • price (real) • total square feet (integer) Include the following **public** methods (*no input/output prompt/display in any methods*): • constructors (default and parameter) • accessor (i.e. getters) methods for all private data members • mutator (i.e. setters) methods for all private data members • method to calculate and return price per square foot using private data members
2	Design the class implementation for a **City** that includes the following **private** data members (*data type in parentheses*): • name (string) • population (integer) • area in square miles (real) Include the following **public** methods (*no input/output prompt/display in any methods*): • constructors (default and parameter) • accessor (i.e. getters) methods for all private data members • mutator (i.e. setters) methods for all private data members • method to calculate and return density (population per square mile) using private data members

CIS 1400: Programming Logic and Technique
Group Project

Team Number	Problem
3	Design the class implementation for a **Meal** that includes the following **private** data members (*data type in parentheses*): • name (string) • cost (real) • number of servings (integer) Include the following **public** methods (*no input/output prompt/display in any methods*): • constructors (default and parameter) • accessor (i.e. getters) methods for all private data members • mutator (i.e. setters) methods for all private data members • method to calculate and return cost per serving using private data members
4	Design the class implementation for a **Gas Vehicle** that includes the following **private** data members (*data type in parentheses*): • type (string) • gallon tank size (real) • miles driven on tank of gas (integer) Include the following **public** methods (*no input/output prompt/display in any methods*): • constructors (default and parameter) • accessor (i.e. getters) methods for all private data members • mutator (i.e. setters) methods for all private data members • method to calculate and return miles per gallon using private data members

CIS 1400: Programming Logic and Technique
Group Project

Team Number	Problem
5	Design the class implementation for an **Employee** that includes the following **private** data members (*data type in parentheses*): • name (string) • pay rate (real) • hours worked (integer) Include the following **public** methods (*no input/output prompt/display in any methods*): • constructors (default and parameter) • accessor (i.e. getters) methods for all private data members • mutator (i.e. setters) methods for all private data members • method to calculate and return total gross pay using private data members