UML Guidelines

- Include UML version on diagram
- Include a header (name/class/assignment #)
- You may omit a simple driver class if doesn't include any real design functionality.
    - Public static void main(String[] args) {
        - App myApp = new App();
        - myApp.run();
    - }
    -
    - If this class contains other attributes or methods it should be included.
- Getters/setters/constructors may be omitted to save space
- Getters/setters should directly match the associated attribute
    - int attr => setAttr/getAttr
- As an industry best practice, most attributes should be private or protected and not public access.
- Your diagram should be able to fit on one page of a pdf, they can be as large as necessary.
    - Ensure diagram is clear when zooming in if it is large
- Relationship lines should be clear. Avoid curved lines. Minimize crossing other relationships and cutting through other classes as much as reasonably possible.
    - If this becomes hard it may indicate a need to refactor your design.
- Access specifiers are not included in lecture videos but they should match your code
    - public: +
    - private: -
    - protected: #
- Class names should begin with capital letters and contain no spaces
    - ClassName not Class Name
- Attributes and methods should not begin with capital letters and contain no spaces
    - Use camel case (newAttr) or underscores (new_attr)
- Return types and parameters for methods should be included, variable names can be omitted to save space. If no return type is listed it is assumed to be void.
    - + setAttr(int) or + setAttr(attr: int)
    - + getAttr() : int
- Cardinalities should appear on both ends of all relationship(s)
- Class and attribute names should be singular.
- Names of Collections can be plural, include the types in your diagram.
    - - courses: Set
    - - heights: int []
- Constants should be capitalized
    - SOME_CONSTANT
- Include types for lists, arrays, sets, etc.
- Attributes / operations should not be on the relationship line or inferred by the relationship
    - They should be listed inside the class model box in the appropriate section.
    - This guidance overrides what is generally considered normal & acceptably by Larman & will be unacceptable for this class.
- Tools that generate UML diagrams from existing code violate the spirit of the assignments and are not permitted.
- No hand drawn diagrams.
- Lucid chart is a good free option and you can sign up for an account using your gatech email address.
- https://www.lucidchart.com/pages/usecase/education (Links to an external site.)
- (Links to an external site.) (Links to an external site.)Draw.io is also another good free option: https://www.draw.io/ (Links to an external site.)
- (Links to an external site.) (Links to an external site.)You are free to use other drawing tools as you see fit.
- https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools (Links to an external site.)
- https://docs.nomagic.com/display/NMDOC/Quick+Reference+Guides?preview=/70389199/70389210/no-magic-quick-reference-guide_uml.pdf (Links to an external site.) (Links to an external site.)
- Tic Tac Toe example: https://github.gatech.edu/GaTechCS6310/tic-tac-toe (Links to an external site.)

This list is not complete, but does provide a good start for some of the points we will be looking at this semester. If you have any others you believe we should add, make sure you let us know.