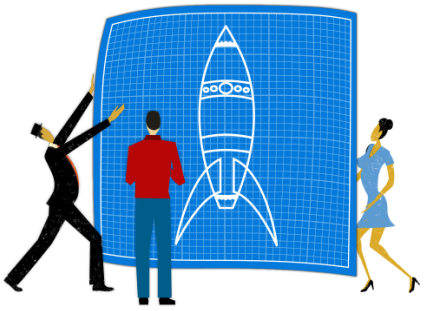
# Class Diagrams Part #1

Re-submit Assignment

**Due** Feb 7 by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip

## Class Diagrams Part #1



This assignment will focus on using UML class diagrams "as a blueprint" for a software artifact. You will model a problem in UML as a group and then implement your design in a Java FX application. Part #2 of this assignment can be found here: **Class Diagrams Part #2**.

## Preparation

- Form groups of three individuals
- Have someone in your group read the problem statement below aloud

## Problem Statement

Jose is a bit of a fitness "nut." Everyone who knows Jose says that he enjoys sports, hiking, and the outdoors. Jose is also an entrepreneur and has a great idea for a new software application. So he hires a new team of software developers to build his dream app, sits down with his team and describes his idea:

*"I have a great idea for a unique Hiking Log application. I want to be able to track my adventures in the outdoors. The application will allow me to plan ahead for my hikes and to track my exercise while I'm out.*

*I'd like to tell the application when I'm going for a hike and where I'm going. I tend to visit the same location often, so being able to type the name of a hike or pick a previous trail name I've entered would be the easiest way to use the program. Once I've told the application I'm going on a hike, I'd want to be able to prepare for the hike. I normally have to find my backpack and pack food and water. I also need to bring practical hiking gear, like a compass, flash light and binoculars. I'd really like to see each of these things as a long list of tasks. I can then check off each task as I complete it.*

*I plan to use my fitbit when hiking and I'd like to also track my fitness details once I've completed my hike. Things like the number of steps I've completed and my average heart rate would be perfect. That way, perhaps in the future, I could see my average steps or improved heart rate.*

*Oh yes, I almost forgot. It would be great to set reminders in the app as well. Messages like "Be sure to stretch!" and "Wear your Fitbit!" could be displayed with each hike. Being able to enter new messages would be important."*

## UML Class Diagrams

Together with your group create a set of class diagrams that represent the software system described above. Your diagrams should include:

- Classes to represent the major "nouns" in the problem description
- Fields representing the attributes of objects
- Methods representing the actions of objects.
  - Only add methods that describe distinctive actions with the class (don't include getters or setters)
- Relationships between classes
  - inheritance
  - association
  - composition
  - aggregation
  - dependency
- Multiplicity of fields (unless they are singular)
- Multiplicity of each relationship
- Name of each relationship

## Group Work Constraints

This project will require you to work in a team environment. Please use the following guidelines when interacting with your team.

- Be patient and respectful with your teammates.
- You must make time to meet with your team. Use any of the following methods:
  - Meeting face-to-face
  - Meeting together using online tools, such as google hangouts
- Team members who do not contribute to a project will not be given credit. Do your part and work hard!

## Requirements

- Complete your class diagrams using the UML tool we discussed in class
- Each of your group members should upload your project files to Canvas
- You will be required to present your design to the class and to justify your design choices
  - Keep an eye on the Canvas calendar for our presentations date

| Class Diagrams Part #1 Rubric | | | |
|---|---|---|---|
| **Criteria** | **Ratings** | | **Pts** |
| Adequate classes are present in your UML diagrams for the Hiking Log application. No major objects from the problem statement are missing from your diagrams. | 15.0 pts Full Marks | 0.0 pts No Marks | 15.0 pts |
| Adequate fields are present in each class diagram describing the attributes of all objects in the problem statement. | 15.0 pts Full Marks | 0.0 pts No Marks | 15.0 pts |
| Each field is given a visibility modifier and multiplicity (unless they are singular). | 10.0 pts Full Marks | 0.0 pts No Marks | 10.0 pts |
| Appropriate methods are defined for each class. Appropriate arguments and visibility modifiers are given. | 15.0 pts Full Marks | 0.0 pts No Marks | 15.0 pts |
| All important relationships between classes are shown using the various notations in UML. All relationships use the correct arrow type according to the UML specification. | 15.0 pts Full Marks | 0.0 pts No Marks | 15.0 pts |
| Each relationship has a name and multiplicities declared. | 10.0 pts Full Marks | 0.0 pts No Marks | 10.0 pts |
| UML diagrams are clean, organized and easy to review. | 10.0 pts Full Marks | 0.0 pts No Marks | 10.0 pts |
| All project diagrams are built using the UML tool we discussed in class. | 10.0 pts Full Marks | 0.0 pts No Marks | 10.0 pts |
| | | | Total Points: 100.0 |