Waverly Hinton
Dominic Fuller-Rowell
Michael Brandt

*1. Implemented Features and final class diagram (This may have changed from what you originally anticipated from earlier submissions)*

Class Diagram: see “Part 4 Class Diagram.png”

We were able to complete the majority of the classes that we had originally attempted. The only class we don’t anticipate completing by thursday would be the RinexMgr.

*2. Did you make use of any design patterns in the implementation of your final prototype? If so, how? If not, where could you make use of design patterns in your system?*

Singleton - GpsPostProcessor
Façade - Parts of GPSTk API accessed through UnitConverter

We used fewer Singletons than in the original design as they seemed unnecessary; the GpsPostProcessor can just instantiate one of each. Adding the Singleton functionality did not provide much more gain.

*3. In addition, the report must discuss how the final system changed from the design you presented in Project Part 2. In particular, include the class diagram you submitted for Project Part 2 and use it to compare and contrast with the class diagram that represents the final state of the system.*

Part 2 Diagram: see “Part 2 Class Diagram.png”

Part 4 Diagram: see “Part 4 Class Diagram.png”

There were a few major differences between our original design and what we were able to complete. The first and most noticeable is that we did not implement singleton class for most of the manager classes that we had. We only had it completed for the GpsPostProcessor class. Secondly, instead of passing an array of doubles for the keeping track of the position, we switched to vectors to make it easier to read. Other than that the changes were minimal and mostly involved renaming methods in classes to make it more obvious what that method is doing. There were some additional private methods added as

well to help dry out our code.

We added to the GPSTk façade section in the UML as we learned more about the GPSTk API and what parts we would need to use.

*4. What have you learned about the process of analysis and design now that you have stepped through the process to create, design and implement a system?*

Planning a project is a time consuming process that can take just as long to actually program it. That being said to try and tackle the same task with no planning would involve insane amounts of refactoring. It's much easier to distribute work when you have discrete tasks that can be integrated together later.