### Concept Inventory

A concept inventory is simply a list of concepts that you go through, indicating how you feel about them. Since all are important, you should make sure you take care with all of them. The concept inventory includes all of the things you should know in some degree by the end of Problem 1 Part A.

Rate each aspect as follows based on how you feel you know it:

1. never heard of it

2. heard of it but know nothing more than that

3. know this well enough to try to apply it

4. know this and can apply it

5. know this well enough to explain it to someone else

|  |  |  |
| --- | --- | --- |
| Concept | Score | Notes on how you may improve your score or on your score |
| Classes and Objects | 5 |  |
| Inheritance | 4 |  |
| Virtual Functions | 3 | I just haven't used virtual functions very often in the programs and would need to find reasons/opportunities to use them – essentially need to write programs with them specifically in mind to use |
| Command Line Arguments | 5 |  |
| Template Function | 4 |  |
| Template Class | 3 | I succeeded in creating templated classes in this project, and it seemed relatively easy – if errors arose I dealt with them sequentially and they were relatively easy to get rid of. I still don't have a strong grasp on it and would just need to utilize them much more in future programs. |
| STL List | 5 |  |
| STL Queue | 5 |  |
| STL Stack | 5 |  |
| Write Own List | 5 |  |
| Write Own Queue | 5 |  |
| Write Own Stack | 5 |  |
| Abstract data types | 4 |  |

* What part of the program are you most proud of and why?
  + I'm fairly proud of the Battleship portion of the program in and of itself, but after leaving the project and then returning back to it I see things that weren't designed as well as they could be and are confusing when attempting to 're-learn' the program. There are errors I didn't see or didn't discover in the first program that I now notice and its not as good as it could be in reflection when I thought it was better.
  + The part that I'm most proud of though is definitely the node/stack/queue which took the longest and then turning them into templated classes – the Project Part B section of the program is really just there to prove that those components were done correctly and the Part B and Part C function the same with both my Queue and the STL Queue.
* What challenges did you face and how did you solve them?

The most challenging section of the program was getting the Queue and the Stack to function properly. It took me by far the longest and essentially set me behind in the class. Templating those classes was actually relatively simple even though it looks less comprehensible syntactically.