The University of Texas at El Paso
Department of Computer Science
CS 3331 – Advanced Object-Oriented Programming
Code Review Checklist

Implementation

- Does this code do what it is supposed to?
  Yes, it does and even a bit more.
- Can it be simplified?
  It can. I ended up merging classes, now there is less classes in the UML class diagram looks much better
- Is the code dynamic or hardcoded?
  It is dynamic, the only thing that is hardcoded is the constant variables.
- Is the code maintainable?
  Now that I have simplified the coded it is maintainable, not adding new functionality requires very little to no change at all.

Logic

- Cases where code does not behave as expected/intended?
  I found that having the iterator created when a Menu object was created was bad since if a new customer was added then the iterator would not of had it.
- Test cases where it may fail?
  The code might fail to produce a correct print statement when dealing with floating points that have more than 2 decimal points after the decimal point

Readability/Style

- Easy to read/understand?
  Somewhat, I feel like some of the documentation is a bit rough. The code is also a bit rough maybe I can change the variable names to make more sense.
- What parts can be modified or adjusted?
  I can merge some of the classes and make the UML class diagram much simpler, this would make the code much more neater and its readability would improve much more.
- Is the structure appropriate?
  I think the structure is fine, I could not find anything that really stood out.
- Does it follow the appropriate language style?
  It does I based my code on what was provided as a reference and I think that it looks ok.
- Is the code well documented?
  I would say yes. I tried to comment anything that might of needed comments to help better understand the code and sometime to even justify why I did something.

Performance

- What is the code complexity?
  The code as a whole is O(infinity) since everything runs in a while loop. But my methods are all O(n) or better, there is nothing that is greater than O(n).

- How does the complexity change with various inputs?
  Of some of my methods yes, since you can search by different inputs such as name and ID, one will be O(1) and the other is O(n) since we iterate over the whole customer collection

Refactoring

- Describe your process for refactoring
  First of all I removed some of the classes and merged the methods withing them to a corresponding classes. Not functionality was changed, everything ended up being the same. I also added a bit more comments just to better help whoever reads the code understand what I was doing. Lastly I ended up making it so that a new iterator is created every time we iterate over a collection, that way outdated iterators are not reused.