

Feedback and Action Plan

Assignment 1 & 2:

System Design:

Tutor's Feedback	Action Plan
A good selection of classes that are drawn from the brief and are consistent with the scenario. The states and behaviours are reasonable for the classes that are included.	Work more on composition and inheritance
The use of composition and inheritance is quite good, but there are several occasions where composition would have been useful, for example, an order containing a customer.	
The states and transitions are good and drawn from the scenario and you have done a good job of capturing the various states of an order.	
A reasonable attempt at a sequence diagram but potentially lacking some of the key ideas. The aim of a sequence diagram is to help understand the way that messages are passed between objects. There are a lot of occasions where you are gathering input from the actor, which is not necessarily what we are aiming to show here – that is perhaps better suited to an activity diagram.	Re-check notes on the sequence diagram and re-work
A good attempt at an activity diagram that captures the logical flow of placing an order. The diagram is well laid out and provides a good deal of information.	Rework start and end points
The one note, I would suggest is that the start and end points are usually represented by filled circles.	
A good overview of your choices in the narrative	

System Implementation

Comments	Tutor's Comments	Action Plan
Readability	<p>The code is generally okay, but difficult to read at times. The Python coding standards (and naming conventions) have not always been followed. For example, variable names should have words separated by an underscore.</p> <p>Try to use whitespace to make your code more readable and accessible to the next developer (or me!). Everything is quite bunched together making it difficult to pick out the key elements.</p>	<p>Re-check naming conventions and re-work</p> <p>Add underscore for variable names</p> <p>Use whitespace</p>
Documentation	<p>The code has some comments, but these are not consistent throughout, with some sections not really explained.</p> <p>The associated documentation is good and provides some insight into the process.</p>	<p>More comments in codelines</p>
Object orientation	<p>A good attempt to leverage the relevant features of object oriented design, including inheritance.</p>	
Code elegance	<p>The code is a little difficult to follow at times, and there is</p>	

	some space for refactoring to improve both its efficiency and readability.	
Meeting the requirements	The system that was developed has done a good job of implementing the design and meeting the requirements.	
Database implementation	The database is very sparse and only contains a single table. This seems like it could be slightly underdeveloped and could have been refined to better fit into the required normal forms. As it stands the database does not comply to the rules of normalisation as required – for example, a delivery address does not only rely on the customer ID.	<p>Since I spent too much time learning SQL on mySQL workbench and understanding Flask, I did not add more tables.</p> <p>-recheck normalisation</p>