



Object Oriented Programming

Distinction Task 6.3: Custom Program

Overview

At this stage you should have enough understanding of object oriented programming to start thinking about creating your own custom program.

Purpose: Demonstrate that you can design and implement your own program using object oriented programming tools.

Task: Create your own program, UML class diagrams, and design document.

Time: This task should be completed before the end of the semester, but progress should be submitted periodically.

Resources:

Note: Only start this once you have your program plan checked by your tutor.

Submission Details

You must submit the following files to Doubtfire (periodically):

- The code for your program (concatenate multiple files into a single file for submission)
- A picture of your UML diagrams — class and sequence diagrams (photo or scan)
- A short design / usage document outlining what your program does and how it works.

Instructions

You have now completed tasks related to all of the unit learning outcomes, and can work toward demonstrating these in your own program. If you are aiming for a Distinction or higher grade you should start working on this program now. Aim to create something of at least the complexity of the drawing and spell/rover programs. Specifically it should:

1. Demonstrate the use of abstraction — create your own classes that model the domain.
2. Demonstrate the use of inheritance and polymorphism
3. Demonstrate the use of UML class and sequence diagrams to explain how your solution works.
4. Demonstrate appropriate use C# coding conventions - case, indentation
5. Demonstrate the use of XML code documentation.
6. Demonstrate appropriate use of structured programming principles (no goto in OO programs either!)
7. Use the checklist on the next page to make sure you have everything you need to submit!

Here are some steps to get you started:

1. Think about what you want the program to do. Maybe write up a paragraph or two to explain it to others. Drawing a picture of what you want it to look like is also a great idea.
2. Show your plans to your tutor, lecturer, help desk staffers, and/or friends to get some feedback.
3. Start thinking about the objects you will need?

Tip: Start small and iteratively build up your larger program!

4. Get something working quickly. You want to see it running ASAP. Once it is working build it a little at a time, get one thing working then move on to the next aspect.

You should periodically submit your work to be checked by your tutor. They can then let you know if you have done enough to meet the Distinction (and High Distinction) criteria.

Note: Your program should be different from the Pass and Credit task programs and from the lecture demonstration programs. You want to demonstrate that you have learnt from these tasks and can apply what you have learnt to some other program design.

If you are aiming for a High Distinction, review the related High Distinction Project document for details on how you can ensure this program meets the HD requirements.

Custom Program Checklist

- ☐ Make sure you have your program plan checked by your tutor. See the associated credit task for details. Ideally this should be signed off before you start writing the code.
- ☐ Ensure that you have reviewed the Distinction and High Distinction criteria from the relevant tasks.
- ☐ Implement enough of the program to demonstrate unit learning outcomes.
- ☐ Create a screencast of your program (video) and upload to an online service.
- ☐ Create a screenshot of your program (image)
- ☐ Create an updated your design report
 - ☐ List the records and enumerations you ended up created
 - ☐ Describe the main functions and procedures in your code. — just the ones that are core to understanding how your program works.
 - ☐ Update your structure chart
 - ☐ Provide a link to your video
- ☐ Submit the design report, screenshot, UML class diagram, UML sequence diagrams, and concatenated code (combine it all into one file for submission).