Faculty of Science, Engineering and Technology

Object Oriented Programming

High Distinction Task 6.4: Custom Program HD Requirements

Overview

To be eligible for a High Distinction grade you must demonstrate that you can use the skills you have learnt to create high quality software solutions that demonstrate the highest programming and design standards.

Purpose:	Demonstrate that you can create high quality software.
Task:	Tidy and extend your custom program, demonstrating that you can create high quality software.
Time:	This must be completed before you submit your portfolio, but it is advisable to submit progress for feedback at earlier stages.
Resources:	▪ Stack overflow ▪ Search engines

Note: If you are not currently up to date you should skip this task and return to it once you are up to date with the Distinction Tasks. Do not allow High Distinction Tasks to delay you in keeping up with the unit's work.

Submission Details and Assessment Criteria

You do not need to submit anything extra for this task.

SWINBURNE UNIVERSITY OF TECHNOLOGY

Instructions

Demonstrate that you can design programs and implement them to a very high standard.

It is recommended that you do this by ensuring that your Custom Program for Distinction meets both the Distinction and High Distinction Standards. However, you can design and implement a second program to meet these standards if desired.

Tip: Show your program to tutors and ask "How can I make this better?". This task is more about quality than it is about quantity.

Your program must demonstrate the following:

- Ability to design and implement a program of reasonable complexity.
  - Program does more than have the user respond to random actions, or simply manipulate data.
  - The program must demonstrate the need to think about its structure and implementation.
- Effective use of object oriented design concepts.
  - The program consists of classes that represent good abstractions.
  - Classes should have small, targeted, methods with little code duplication.
  - Methods perform meaningful actions.
  - Data and objects are used intelligently to minimise the amount of code required.
  - Each class is conceptually coherent with good cohesion and appropriate coupling.
- Use of good programming practices.
  - Code is correctly indented, with meaningful names assigned to all identifiers
  - Code is commented to help the reader understand the abstractions and how they work.

Tip: Consider adding things like levels to a game, or multiplayer support. Adding things like networking or using files to store program data can help you increase the complexity of a program.

Tip: Keep a journal of the programming ideas you have and notes on your design decisions. These can help you explain how your program meets these criteria in your portfolio.