Introduction to Programming

High Distinction Task 9.3: Custom Program High Distinction 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 and use other APIs.
Task:	Tidy and extend your custom program, demonstrating that you can create high quality software. **You must meet the HD custom program requirements – Please check the rubrics for this on Blackboard**. Your grade will depend on which level in the Rubric you demonstrate.
Time:	This must be completed before you submit your portfolio, but it is advisable to submit progress for feedback at earlier stages.
Resources:	Freider, O, Frieder, G & Grossman, D. 2013 *Computer Science Programming Basics in Ruby*, O'Reilly Media _Available online from the Swinburne Library_.

- 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 Credit and Distinction Tasks. Do not allow High Distinction Tasks to delay you in keeping up with the unit's work.

Submission Details

You must submit the following files to Doubtfire:

- Program source code, and screenshot of the program in action.
- Design Report (for the Custom Code design task)
- **NB: Code must be written as Structured Code (i.e using functions and procedures, not Object Oriented design) you can however use Object Oriented libraries (gems), such as Gosu.**
- **You must be able to explain your code.**

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 lecturers 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 (eg. Food Hunter), or simply manipulate data (eg. Sort Visualiser, or the array of records program).

  - The program must demonstrate the need to think about its structure and implementation.

- Effective use of functional decomposition.

  - The program consists of small functions and procedures with little code duplication.

  - Data is used intelligently to minimise the amount of code required.

- Effective use of abstraction.

  - Functions and procedures represent meaningful tasks.

  - Data types are used to model entities associated with the program

- 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 features (eg: networking) or using data cleverly to customize and drive your program (eg: state machines) 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. **It is important to talk about the problems you encountered and how you dealt with them. Do this in your video for your interview. Doing this well is necessary for the higher grades.**