Faculty of Science, Engineering and Technology

Introduction to Programming

Distinction Task 2.1: Custom Program

Overview

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

Purpose:	Demonstrate that you can design and implement your own program using structured procedural programming tools.
Task:	Create your own program, structure chart, and design document.
Time:	This task should be completed before the end of the semester, but progress should be submitted periodically.
Resources:	▪ Programming Arcana ▪ Swinburne CodeCasts (YouTube Channel, iTunesU) • Making the most of the concept of abstraction

> **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 short design / usage document outlining what your program does and how it works.

▪ A picture of your structure chart (photo or scan)

SWiN
BUR
* NE *

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 Food Hunter program for the lower distinction grade or more complex for higher grades. Specifically it should:

1. Demonstrate the use of functional decomposition - implement the program with a number of functions and procedures. (Maybe even modular decomposition with separate units if you can identify some reusable artefacts - optional but nice)

2. Demonstrate the use of arrays and records

3. Demonstrate the use of structured programming (sequence, selection, and repetition)

4. Demonstrate appropriate use coding conventions - case, indentation

5. It must not use global variables, or goto.

6. 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 data - what records and enumerations will you need?

> **Tip**: Start small, you can easily add to records at a later stage. Try to identify what records you will need, then add just the basic data - enough to get something working. Once that first part is working, add additional fields as they are needed.

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 food hunter program and 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 and check the marking rubrics 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 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

☐ Submit the design report, structure chart, and concatenated code (combine it all into one file for submission).

☐ Start producing your video (see Distinction Task 2.2)