Faculty of Science, Engineering and Technology

Object Oriented Programming

Topic 10: Exceptions

Resources

The following resources can help you with this topic:

- www.cplusplus.com Tutorials and Forums
  - Dynamic Memory
  - Special Members (including copy constructor)
- Tutorials Point C++ Programming Tutorials
- *Learning Material / Content / Outline and Assessment / Research Topics and Report*

Topic Tasks

Before starting to work on the tasks, first read through this entire document to get a sense of the direction in which you are heading. Complete the following tasks and submit your work to Doubtfire for feedback before the deadline (check Doubtfire for the submission deadline).

Credit Task 4 - Case Study: Iteration 7 2

Distinction Task 3 - Custom Program Unit Tests 3

High Distinction Task - Plan Research 4

Remember to submit your progress, even if you haven’t finished everything.

After you have **discussed** your work with your tutor and **corrected any issues**,
it will be signed off as complete.

SWINBURNE UNIVERSITY OF TECHNOLOGY

Credit Task 4 - Case Study: Iteration 7

Complete Iteration 7 - Paths and Moving

Once your tests are working correctly add additional details to your cover page. Remember to relate what you are doing to the unit's learning outcomes.

> **Tip**: Now would be a good point to reflect on anything you did in these iterations that was interesting or challenging. Elaborate on this in the cover page so that you can take advantage of this in your portfolio.

Credit Task 4 - Assessment Criteria

Make sure that your task has the following in your submission:

- Iterations 1 to 7 are implemented.
- Class and Sequence Diagrams are included to illustrate additions to the design.
- The new classes have XML documentation for all public features.
- Code must follow the C# coding convention used in the unit (layout, and use of case).
- The code must compile and the screenshot must show the tests passing.

Distinction Task 3 - Custom Program Unit Tests

If you are aiming for a Distinction or High Distinction grade, you should demonstrate weekly progress on your custom to your tutor. This will help to ensure that your custom program will meet the Distinction criteria, and your tutor should be able to give you advice on how to structure your program.

This week, choose the most abstract class in one of the inheritance hierarchies in your design and write a set of unit tests that any child of this class must pass, then demonstrate that all derived classes pass this test.

> **Tip**: Inheritance is one of the strongest forms of coupling between classes, and changes to a base class are risky because they have a tendency to cascade down through all of the children. Therefore, unit tests of base classes are an important way to verify that modifications have not broken any of the original assumptions.

Distinction Task 3 - Assessment Criteria

Include the following in your submission:

- Unit tests that verify the correct behaviour of different children via the parent class interface.
- Well documented code that demonstrates careful attention to standard conventions.
- Screenshot that shows the tests passing.

High Distinction Task - Plan Research

Note: Start discussing your research project with the lecture and tutor as soon as possible but do not start working on it until after you have completed your custom program

Create an outline of the research question and method for your research project.

> The **research question** is the question you aim to investigate in the project.
>
> The **research method** describes how you will approach answering the question.

Tip: Discuss both the question and the method with your tutor and lecturer before starting the project.

To achieve a High Distinction result you need to conduct a small research project. This project will seek to answer a question related to the unit. For example, "What is more efficient, an array or linked list?". See the HD Research Ideas document for further examples.

> A **satisfactory** research report will provide data to support the answers given.
>
> A **good** research report will expand upon the data to discuss *reasons* behind the results identified.

Hint: You want to demonstrate your ability to **analyse** and **explain** findings. Your reasons should link what you have learned back to programming in general. You should make sure that you explain any cases where results differed from your expectations.

Tip: Keep the focus of the research small, you do not have very much time to find the answers to larger questions. For the arrays vs. linked list example, you would want to focus on just a few different tasks.