

Notes for the Teacher

Each exercise has five sections:

1. the basic problem (all of these are listed in the document for students, exercises.doc)
2. hints to assist students who are unsure about how to proceed
3. a sample answer
4. some exercise specific marking guidelines
5. suggested extensions to the initial problem for more able students

The exercises are roughly ordered in increasing difficulty, so it is advised that students work through them in numerical order. Also although the exercises don't follow on directly, some of the later questions can utilise concepts and code from the earlier exercises.

The main focus of the exercises is on the use and debugging of pointer manipulation, dynamic memory, and structured data types, as it was felt that these were the most challenging and complex topics in the basic C / C++ language, without getting into platform specific areas such as graphics and windowing systems or focusing heavily on OO concepts.

Use of library functions should be kept to a minimum of I/O routines and memory allocation. Although there are standard library functions which have the effects required for solving some of the exercises, students are encouraged to create alternatives. Standard library functions are by their nature comparatively simple to use, so aren't really appropriate for exercises. Students submitting solutions that are wrappers for library functions should be marked down correspondingly.

The solutions provided are by no means the only method of completing the exercise, especially with the later questions there are several radically different approaches that are possible, students should be given additional credit for finding more optimized solutions.

Comments are included in the solution as general good practice and to explain the solution. All answers should include error handling to cover as many possibilities and invalid inputs as possible. All exercises are geared towards console applications as opposed to windows based systems.

Marking

Each of the exercises is marked out of 20, with the marks being awarded as follows:

Category	Marks	Criteria
Commenting	0	Not comments present.
	1	A few scattered comments.
	2	Solution is well commented.
Spacing	0	Lack of spacing reduces readability.
	1	Code is practically spaced.
Indentation	0	No / erratic indentation.
	1	Consistent indentation.
Style & Efficiency	4	The marker's discretion on the quality and efficiency of the code submitted
Exercise	8	Dependant on exercise
Compilation	0	The solution doesn't compile.
	2	The solution compiles.
Functionality	0	Program doesn't work.
	1	Program performs some of the required tasks.
	2	Program performs all required tasks.