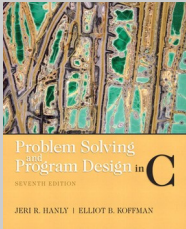
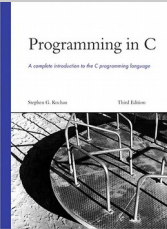
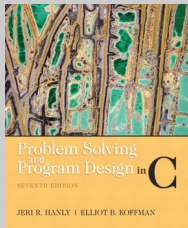
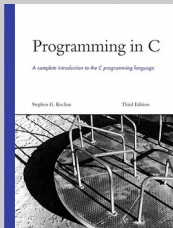
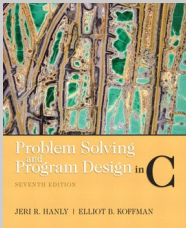
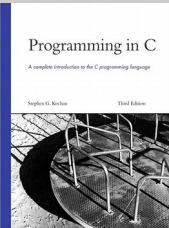


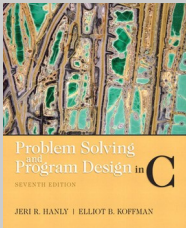
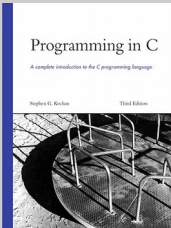
UNIX & C Programming (COMP1000) – Weekly Reading (and Textbook Advice)

Any C book is better than none, and others exist besides the three mentioned here. Seek advice from the lecturer/unit coordinator on their usefulness.

	 <p>Hanly and Koffman (2013), <i>Problem Solving and Program Design in C,</i> 7th edition. (Recommended)</p>	 <p>Kochan (2005), <i>Programming in C: A Complete Introduction to the C Programming Language,</i> 3rd edition.</p>	<p>Banahan, Brady and Doran (1991), <i>The C Book.</i></p>
Textbook overview	<p>This book introduces pointers <i>before</i> arrays (unlike other C books). I believe this is the best approach, and structure of the unit reflects this.</p> <p>This book isn't perfect, but overall is preferred to the others.</p>	<p>Has been used in Engineering Programming (COMP1004) and previously in this unit.</p> <p>Unfortunately, this book (1) <i>does not</i> cover pointers and dynamic memory allocation in detail, and (2) uses C99 constructs that don't work in C89. However, it <i>does</i> have a reasonably good chapter on debugging, unlike the others.</p>	<p>Freely-available online. The age of this book is not really a problem. It covers the same C standard (C89) used in this unit.</p> <p>This book has a more technical style than the others, and is suited to those who are relatively confident with programming in general.</p>
Prerequisite knowledge from OOPD 110 or EP 100.	<p>Chapter 1: Overview of Computers and Programming Chapter 4: Selection Structures: if and switch Statements Chapter 5: Repetition and Loop Statements (except for Section 5.11, which you can ignore).</p>	<p>Chapter 2: Some Fundamentals Chapter 4: Variables, Data Types, and Arithmetic Expressions Chapter 5: Program Looping Chapter 6: Making Decisions</p>	<p>Chapter 2: Variables and Arithmetic Chapter 3: Control of Flow and Logical Expressions</p>
Lecture/Prac 1: Revision	<p>Chapter 2: Overview of C Chapter 3: Top-Down Design with Functions</p> <p>Much of this is <i>also</i> prerequisite knowledge, but you should go over it quickly at least, especially if you come from Object Oriented Program Design (COMP1001) and haven't seen C before.</p> <p>Section 3.6 deals with graphics, which is interesting but beyond the scope of the unit (and may be difficult to get working on the lab machines).</p>	<p>Chapter 3: Compiling and Running your First Program</p> <p>Go over this quickly if you haven't seen C before.</p> <p>Chapter 8: Working with Functions</p> <p>Ignore the parts on arrays (for now), and the section on recursive functions (which is beyond the scope of the unit).</p>	<p>Chapter 1: An Introduction to C Chapter 4: Functions</p> <p>Section 4.3.3 (Recursion) is beyond the scope of the unit. Section 4.4 (Linkage) is covered briefly in lecture 2.</p>

	 <p>Hanly and Koffman (2013), <i>Problem Solving and Program Design in C</i>, 7th edition. (Recommended)</p>	 <p>Kochan (2005), <i>Programming in C: A Complete Introduction to the C Programming Language</i>, 3rd edition.</p>	<p>Banahan, Brady and Doran (1991), <i>The C Book</i>.</p>
Lecture/Prac 2: Environments	<p>Chapter 12: Programming in the Large</p> <p>This chapter is useful here for its discussion of macros and header files. Unfortunately, some of the examples use structs, which are not covered until lecture 6. Understanding structs is not crucial until then.</p>	<p>Chapter 13: The Preprocessor Chapter 15: Working with Larger Programs</p> <p>Chapter 15 mention the make command, but not in detail.</p> <p>Appendix C: Compiling Programs with gcc.</p>	<p>Section 4.4: Linkage Chapter 7: The Preprocessor</p>
Lecture/Prac 3: Pointers	<p>Chapter 6: Pointers and Modular Programming</p> <p>Section 6.1 <i>also</i> introduces “Pointers to Files”, which you can safely ignore until lecture 5.</p> <p>Appendix A: More About Pointers</p> <p>This is important material, even though it's in an appendix.</p>	<p>The following sections from Chapter 11:</p> <ul style="list-style-type: none"> • Defining a Pointer Variable • Using Pointers in Expressions • Pointers and Functions • Operations on Pointers • Pointers to Functions • Pointers and Memory Addresses <p>Also, Dynamic Memory Allocation in Chapter 17.</p> <p>For now, skip over the parts in Chapter 11 related to arrays and structs.</p>	<p>Chapter 5: Arrays and Pointers Section 8.3: Typedef Section 9.15.3: Memory allocation</p> <p>This book, like Kochan, tries to teach arrays <i>before</i> pointers, and as a result it's very difficult to isolate the discussion of pointers.</p>
Test 1 – revise everything up to here			
Lecture/Prac 4: Arrays and Strings	<p>Chapter 7: Arrays Chapter 8: Strings</p> <p>Chapter 7 introduces the const keyword and enumerated types. Both are fairly simple, but are not covered in the lectures until lecture 9.</p> <p>Section 7.10 deals with graphics again, which you can ignore completely.</p>	<p>Chapter 7: Working with Arrays</p> <p>Has a section on “The const qualifier”, which we don't cover until lecture 9.</p> <p>Pointers and Arrays in Chapter 11. Chapter 10: Character Strings Command-Line Arguments in Chapter 17.</p>	<p>Chapter 5: Arrays and Pointers (continued from before) Section 9.16: String handling Section 9.15.1: String conversion functions Section 10.2: Arguments to main</p>

	 <p>Hanly and Koffman (2013), <i>Problem Solving and Program Design in C,</i> 7th edition. (Recommended)</p>	 <p>Kochan (2005), <i>Programming in C: A Complete Introduction to the C Programming Language,</i> 3rd edition.</p>	<p>Banahan, Brady and Doran (1991), <i>The C Book.</i></p>
Lecture/Prac 5: Input and Output	<p>Chapter 11: Text and Binary File Processing</p> <p>Consider reading chapters 10 and 11 together, or at least briefly look at structs before reading chapter 11.</p>	<p>Chapter 16: Input and Output Operations in C</p>	<p>Section 9.10: Input and output Section 9.11: Formatted I/O Section 9.12: Character I/O Section 9.13: Unformatted I/O Section 9.14: Random access functions</p>
Lecture/Prac 6: Structs	<p>Chapter 10: Structure and Union Types Sections 13.1 to 13.4 in Chapter 13</p> <p>Unions (Section 10.6) are not crucial until lecture 9.</p> <p>Dynamic data structures <i>other</i> than linked lists (Sections 13.5 onwards) are beyond the scope of the unit.</p>	<p>Chapter 9: Working with Structures Working with Pointers and Structures, in Chapter 11</p> <p>This book uses structs in somewhat unconventional ways. The examples of linked lists are not very useful.</p>	<p>Chapter 6 (Structured Data Types), up to and including Section 6.2.2.</p> <p>This chapter covers a large range of topics. We're only interested in structs and linked lists here.</p>
Test 2 – revise everything up to here			
Lecture/Prac 7: Shell Scripting	Not covered in this book.	Not covered in this book.	Not covered in this book.
Lecture/Prac 8: Debugging and Testing	<p>Section 5.10: How to Debug and Test Programs Section 6.7: Debugging and Testing a Program System</p> <p>Both sections are quite brief. Unfortunately, gdb itself is not covered in this book.</p>	<p>Chapter 18: Debugging Programs</p>	Not covered in this book.
Lecture/Prac 9: Miscellaneous C	<p>Section 7.7 covers enumerated types. Section 10.6 covers unions. Appendix C covers bitwise operators. Other concepts in this chapter aren't really discussed in the book, but they are not difficult to grasp.</p>	<p>The const Qualifier, in Chapter 7. The Keyword const and Pointers, in Chapter 11. The volatile Qualifier, in Chapter 17. Working with Unions, in Chapter 17. Chapter 12: Operations on Bits Chapter 13: More on Data Types</p>	<p>Chapter 6, from Section 6.3 onwards: Unions, Bitfields, Enums, etc. Section 8.4: Const and Volatile</p>
Exam – revise everything up to here			

	 <p>Hanly and Koffman (2013), <i>Problem Solving and Program Design in C,</i> 7th edition. (Recommended)</p>	 <p>Kochan (2005), <i>Programming in C: A Complete Introduction to the C Programming Language,</i> 3rd edition.</p>	<p>Banahan, Brady and Doran (1991), <i>The C Book.</i></p>
Lecture 10: C++ (not assessed)	<p>Chapter 15: On to C++</p> <p>C++ is not assessed in this unit, but it is worth playing with in your own time.</p>	<p>Chapter 19: Object-Oriented Programming</p> <p>Fairly brief overview of OOP that includes C++, but also Objective-C and C#.</p>	Not covered in this book.
Extra (beyond the scope of the unit)	<p>Chapter 9: Recursion</p> <p>Recursion is covered in Data Structures & Algorithms (COMP1002). (However, it may be of interest when designing linked list-related algorithms.)</p>	<p>Recursive Functions in Chapter 8.</p>	<p>Section 4.3.3: Recursion</p>
	<p>Chapter 13: Dynamic Data Structures, Sections 13.5 onwards.</p> <p>Stacks, queues and trees are also covered in Data Structures & Algorithms.</p>	Not covered in this book.	<p>Section 6.2.3: Trees</p> <p>Contains a brief example of a binary tree in C. Other data structures like stacks and queues are not covered.</p>
	<p>Chapter 14: Multiprocessing Using Processes and Threads</p> <p>Discussed in UNIX Systems Programming (COMP2002).</p>	Not covered in this book.	Not covered in this book.