Table 1.1: The chapters in the book with a brief summary of each

Chapter	*Contents*
1	Introduction to the system life cycle, object-oriented development and the bike hire case study.
2	Requirements for the new Wheels system, using standard requirements techniques, such as interview, questionnaire and scenario. This chapter ends with a list of requirements for the system that is developed in the rest of the book.
3	The technique of use cases and how the system will interact with its users.
4	Objects, classes and the central ideas in object-oriented development.
5	How to construct a class diagram.
6	Class responsibilities and CRC cards. Sequence and collaboration (interaction) diagrams, showing how all the objects involved behave during a single use case.
7	State diagrams, showing how the different objects of a single class behave through all the use cases in which the class is involved.
8	Activity diagrams, providing details of the activities that take place during a system process.
9	Design at the overall system level, including designing the overall architecture of the system, selecting a strategy for coping with persistent data and designing the user interface.
10	Design at the detailed system level, showing how early models of the system are refined and enhanced as development moves towards program code.
11	The code for the Wheels bike hire system, illustrating the relationship between it and some of the models that have been produced during development.

develop a software system. Building systems is a skill – like playing a musical instrument or swimming – that can only be learnt by practice. It's no good knowing all about swimming if you can't actually swim, and it's no good knowing all about how to develop systems if you can't actually do it. If you simply want to be able to talk about development of software systems in theory, then all you need to do is read the book, but if you want to be able to make a