Preface

Hello and welcome to the unit Object-Oriented Development. In this unit we examine the most popular paradigm for software development in use in the IT industry, object-oriented development. This workbook is intended to guide you through your studies for this trimester and to provide introductory coverage of all the unit content. Please start by reading this section which highlights the many resources for this unit and advises you how to use the workbook.

CloudDeakin

Before starting, it is important to first identify all of the relevant materials for this unit and confirm how to access those materials. The most important resource, outside of this workbook, is the CloudDeakin web site:

> http://www.deakin.edu.au/clouddeakin

You are expected to access this site regularly during the trimester (at least twice per week) to check for updates and announcements for your units. To log in to CloudDeakin you will need your Deakin user name and password which are used to access many IT services provided by Deakin. Further information on obtaining your user name and password can be found on the IT Support web site:

> http://www.deakin.edu.au/about-deakin/administrative-divisions/esolutions/it-help

Please take a few minutes to familiarise yourself with the resources available on CloudDeakin, as you will be using this web site many times throughout the trimester.

You should begin by reading the Unit Guide in CloudDeakin, which includes the following useful information:

- A list of resources for this unit, and where to find them;
- Staff involved in the unit and how to contact them; and
- Assessment information including assignments and due dates, specimen solutions (after all assignments are submitted), marks, and how your final mark is calculated.

Prescribed textbook

The prescribed textbook for this unit, at time of printing, is as follows:

- Deitel, P.J., and Deitel, H.M., *Visual C# 2012: How to Program*, Fifth Edition, Prentice-Hall, 2014.

This is a very detailed textbook that you will find important during the trimester, however there are additional resources (additional chapters, source code downloads, etc.) available via a companion web site. Information on accessing the companion

web site, including the required access code, can be found inside the front cover of the textbook. The majority of information required for this unit can also be found in older editions of the textbook, however note that older editions will be missing some information introduced in the newer versions of C#. Also note that purchasing a second hand book will likely mean you will be unable to access the companion web site as the activation code will most likely have been used.

Other textbooks

Although the prescribed textbook provides an excellent reference book for the unit and includes the prescribed readings, some students may find the use of other supplemental textbooks useful when learning C# and/or object-oriented development. There are many textbooks available covering the fields of object-oriented analysis and design, UML, and the C# programming language, each of which may be useful during your studies in this unit. For the first two fields of study, we recommend the following two textbooks:

- Blaha, M., and Rumbaugh, J., *Object-Oriented Modeling and Design with UML*, Second Edition, Prentice-Hall, 2005. (also available through Safari Books Online, see below).
- Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modelling Language*, Third Edition, Prentice-Hall, 2004.

There are also many textbooks available covering the C# programming language, however the textbook that would be most relevant for this unit would be:

- Schildt, H., *C# 4.0: The Complete Reference*, McGraw-Hill, 2010.[1]

Please note that the textbooks mentioned in this section are not required to be successful in studying this unit, however they may provide further reading and/or clarification of some topics. Also note that in general the language and terminology may vary somewhat between different textbooks. In this unit we aim to keep the use of terminology consistent with the prescribed textbook, but will try to point out any popular alternative terms that may be used in other texts.

Safari Books Online

The Deakin Library maintains a subscription to the Safari Online Books service, which provides an online edition of a number of textbooks. It is strongly recommended that all students familiarise themselves with the Safari Books Online service as you will find the service particularly useful when completing almost any IT unit at Deakin. Instructions for accessing Safari Books Online can be found in CloudDeakin.

Software Requirements

[1] Note that the unit SIT102 Introduction to Programming has prescribed a similar textbook by the same author over the years, *C#: A Beginner's Guide*, an introductory textbook containing a subset of the content found in this text.

To complete your studies of this unit, you will need access to software for (i) developing applications using the C# programming language; and (ii) preparing UML diagrams as part of a software specification/design. Information on obtaining this software is provided in CloudDeakin.

How to use this workbook

This workbook is divided into 11 chapters, each matching one week of study in the Deakin trimester. Each chapter is divided into the following sub-sections:

i. Introduction and Objectives – this section briefly introduces the topics for the week, highlighting areas that you should pay particular attention to, and what you are expected achieve by completing the chapter;
ii. Prescribed Readings – a list of sections from the prescribed textbook that you should read in conjunction with this workbook;
iii. Main text – description of the concepts that you need to learn for the particular session;
iv. Exercises – tasks to be completed as part of your studies (during practical classes for on-campus students), solutions for which will be released at the end of each week;
v. Further Problems – optional tasks that you can complete from the textbook if you are looking for more examples; and
vi. Past Exam Questions – a list of questions that have appeared on past exams, to allow you to check your knowledge against the exam requirements (note that parts of questions not covered by a particular chapter are omitted or crossed out).

You will also find a number of resources in CloudDeakin related to this workbook:

- A single PDF file containing the complete workbook;
- Several PDFs containing each section of the workbook separately;
- Full source code for example programs found in the textbook, and where applicable, source code required for some of the exercises; and
- Specimen solutions for exercise tasks (released progressively throughout the trimester).

Reading syntax in this workbook

While reading this workbook, you will come across specifications of language syntax, e.g., the following syntax is presented in Section 1.4 for declaring variables:

type *name1*[= *value*][, *name2*[= *value*][, …]];

The following elements can be seen in these syntax specifications:

- *Changeable elements* (required) – these elements indicate part of the syntax that are required to complete a statement, however different values may appear for each element. In the above example, type must be substituted with a data type supported by C# such as int for an integer variable.

- *Symbolic names* (required) – these elements require the programmer to specify an identifier (a name) for code elements such as variable/attribute names, function names, class names, and so on. In the above example, a programmer creating a variable to store a person's age could substitute name1 for the variable name age.
- `Code elements` (required) – these elements indicate parts of the syntax that must appear directly in the statement in the program code and are formatted using a different font (Courier in this workbook). In the above example, the equals symbol (=), comma (,), and semi-colon (;) symbols must all appear in the final statement (note that commas are inside optional elements however, see below).
- [Optional elements] – optional elements, surrounded by square brackets ([ ]) are used to indicate where the programmer may choose to include in a statement or not. The optional elements indicated in the above syntax indicate that more than one variable can be declared on the same line, and that for each variable declared the programmer can optionally specify the initial value assigned to that variable.
- … – three dots/periods/full-stops (an ellipsis) are used to indicate where a particular pattern of syntax continues to repeat after that shown in the specification. The above example syntax shows that a single variable can be declared, optionally with an initial value, optionally followed by a second variable with optional initial value, followed by an ellipsis. The ellipsis indicates that a third, fourth, fifth, and so on, variables can also be declared by following the same pattern of inserting before the semi-colon a comma, the variable name, and then optionally followed by an equals symbol and initial value.

The following examples all follow the above specification:

```
int age;
int age = 18;
int first, second;
int first = 1, second;
int first, second = 2;
int first = 1, second = 2;
int first, second, third, fourth, fifth = 5;
```

and so on.