

Unit outline

Unit overview

Introduction

The purpose of this unit is to introduce you to the techniques involved in the design of interactive, event driven software. This will be partly theoretical and partly practical. The unit is not essentially designed to make you an expert in any one specific programming language. The language used will be Microsoft Visual Basic but the unit is more than the language.

When we learn to drive a car, we are conscious of learning skills and acquiring knowledge which will be transportable to any car we drive. We do not consider we are 'learning to drive a Toyota' or 'learning to drive a Ford' although we may be using one of these cars as the vehicle in which we are being taught. Similarly, this unit is not essentially about 'learning Visual Basic'. It is about acquiring the knowledge and skills needed to design and implement commercial software. As with learning to drive, you must learn on a particular vehicle—a programming language in this instance. It is necessary, however, to look beyond the language and to regard the knowledge and skills as generic and transportable to other languages.

A framework for learning

As mentioned above, look beyond merely acquiring familiarity with Visual Basic as the aim of this unit.

Specific points to look for as the unit progresses include:

- the nature of event driven software. The software with which most of our personal work is done comprises word processors, spreadsheets and similar products. We often refer to it as the generic term 'Windows-based'. It is software which is highly interactive and responds to our actions with the keyboard or mouse. This is only one model, or paradigm, of software however. Much of the world's commercial data is processed in off-line, batch-processed mode with little or no direct operator intervention. Such software is often referred to as procedure driven. The structure of event driven and object-oriented programs is considerably different from that of procedure driven programs. Visual Basic is only one of a number of similar products facilitating the design of this type of software.
- the nature of prototyping and rapid application development. The first three decades of commercial software development using languages such as PL/I or COBOL were typified by long development cycles extending over many months. The languages offered little support for the rapid development of a prototype to show the prospective users of the system what they might expect from the completed product. Consequently, the delivery of the final product was often accompanied by expressions of disappointment from the users as, only now, were they able to see what they were getting. It is claimed that the advent of products such as Visual Basic, Delphi and other similar development tools has enabled the rapid production of a prototype to become a reality. The user may be presented with a look-alike system for comment and criticism with a fraction of the effort required to build the completed system. Such comments may then be incorporated in the design process and the final product becomes a joint production of users and software developers. You should look to form an impression as to whether or not such a claim is justified.

- the use of an Integrated Development Environment (IDE). Earlier programming languages often comprised little more than a compiler to convert the language's source code to the object code of the computer on which it was to be executed. The programmer was expected to write the program using a text editor supplied, perhaps, as part of the operating system. Debugging tools were non-existent and the programmers were expected to rely on their own ingenuity to detect and correct errors in their programs. Products such as Visual Basic come packaged with features including:
 - an intelligent editor which detects syntax errors as they are written;
 - the facility to execute and test programs without having to first compile them into object code;
 - interactive debugging facilities; and
 - on-line help whereby information is provided on screen without the necessity of consulting voluminous, printed manuals.

Unit structure

As mentioned earlier the unit consists of theoretical material and practical work. The theoretical material is contained within the Study Guide. For on-campus students it will be covered in lectures (one 2-hour lecture each week) and is organised into the following twelve topics:

Topics

- 1 Introduction to software development
- 2 Data definition
- 3 Simple procedure design
- 4 Introduction to sub-procedures
- 5 Program testing
- 6 Design of event driven software
- 7 Introduction to array processing
- 8 File organisation and processing
- 9 Searching and sorting arrays
- 10 Elegance in procedure design
- 11 Program documentation
- 12 Software quality

Within this material we have purposely avoided mention of any specific programming language because we want to emphasise that the concepts behind software development are independent of the actual language used to develop the software. The material in the lectures, Study Guide and the Readings is the only material you will need for the theoretical component of the unit, that is, no textbook is required. You may, however, choose to buy a textbook for the practical component. (See 'Recommended textbook' in the 'Learning resources' section.)

Practical work

Each week, commencing in week 1, there will be **one 1-hour tutorial**.

Usually this will involve two distinct activities:

- The first will be to read sections from the Visual Basic Summary (see ‘Learning materials’ section). The purpose of this reading is to link the theory in the topic to Visual Basic implementation. Not all topics have such readings but when they do there will be a direct correlation between the topic and the prescribed section in the Visual Basic Summary.
- The other task that you will be directed to complete is a set tutorial. The practical skills you need to acquire in this unit will be developed by completing a weekly tutorial and two assignments. In the tutorial work you will be guided through the steps needed to complete two small applications. These applications are similar in size and scope to your assignments and will provide you with the skills you need to complete your assignment work. We recommend that after you complete a tutorial you attempt to apply the same skills to your assignment. In this way, you will develop your assignment in parallel with the tutorial work and therefore be assured of completing the assignments on time. Although the tutorials demonstrate the principles of software development there may not always be a direct relationship between a topic and a tutorial. Again this is an educational decision as we believe it is more important for you to develop a complete working application rather than a series of exercises designed to illustrate particular topics.

Also, from week two the tutorials will include paper and pen pseudocode exercises designed to help you develop skills in program design. You will be required to work through these exercises and submit them for assessment as requested.

Most students find developing applications in Visual Basic to be very enjoyable. However, it is important to remember that this unit is about the software development process. It is **not** a Visual Basic unit. You will be examined on the material in the Study Guide and covered in lectures, as well as your ability to solve problems in code, that is, the skills you develop by completing the pseudocode exercises at the end of the practical work each week.

On-campus students please note: It is unlikely that the practical work set can be completed in the one hour of supervised time each week. We expect therefore that students will download the tutorial **several days prior to their scheduled class**, and commence the tutorial in their own time. The scheduled one hour period, with a tutor present, should then be used to submit your completed pen and paper pseudocode exercises for assessment, collect your marked exercises from the previous week, demonstrate your programming work to the tutor, clarify any problems you have had, and complete any remaining components. Students should make sure that they complete all tutorial work before the start of the next week to avoid falling behind. If you have questions between your scheduled weekly one-hour tutorials you can avail yourselves of the additional help given by the **duty programmer**. Your lecturer will inform you of the time/location of the duty programmer service.

Tutorials for off-campus students

Off-campus students will be organised into tutorial groups in FirstClass after they have signed on for tutorials. To ‘sign on’ leave a simple message in the sign on conference in the MSC211 off-campus tutorial conference. You will then be allocated to a tutorial group conference. After you have been allocated to a tutorial group, a new conference folder labelled “MSC211 Group x” will appear on your FirstClass desktop, where x is a letter of the alphabet.

Pseudocode group tutorials

Only the pseudocode group tutorial exercises will be assessed. We cannot individually assess pseudocode exercises for off-campus students. A **group submission** is required each week. Each week one person in the group is to provide an initial pseudocode solution. During the next week other members of the group are to comment on the proposed solution. After group agreement is reached, the person responsible for the initial solution must submit the agreed solution on behalf of the group. It is expected that the provider of the initial solution will be rotated among the group.

Ten per cent of your assessment will be made up of your submissions and comments on work submitted by other members of your group.

The practical Visual Basic tutorial exercises might or might not run as required. They do not need to be assessed by staff. You are advised to keep up with these exercises because skills developed in doing these exercises will be needed for your assignments.

Student commitment

Students are expected to spend a minimum of eight (8) hours per week, in addition to lecture and tutorial time, on review of lecture/Study Guide and tutorial material, assignment work and study for the end of semester examination.

Student consultation

At some point you may need to consult with your tutor/unit coordinator outside of your tutorial time. Please note sessional tutors are not available for consultation outside of tutorial time. Staff tutors will have set student consultation times. These times are listed on the tutor's door. All students (from any tutorial) are welcome to consult with any tutor during set student consultation times. Off-campus students should use the FirstClass conferencing facility.

Conclusion

The study of this unit will be rewarding in that it will give you an insight into the processes involved in the design and implementation of commercial software, it will provide you with extensible and transportable knowledge and skill in developing such software and, as the result of the assignments, it will leave you with small applications which will have the look-and-feel of professional software. It will also develop your general skills in solving problems. The common thread running through all software development is that of problem solving. In order to write software one must be able to analyse a problem, break that problem down into smaller, manageable segments and then combine those segments into a correctly functioning whole. These are skills which are generic and enhance our ability to cope with all aspects of our work and much of our personal life as well.

Learning resources

You will have access to several sources of information about the material comprising this unit.

On-campus students

Provided by Deakin University

- One 2-hour lecture each week (see timetables)
- One 1-hour tutorial each week (see timetables). Register using Etute (see website below for details)

- Access to duty programmer support (your lecturer will advise of time/location)
- Copies of the weekly lecture notes and tutorial exercises via FirstClass

The various notes are in Acrobat format (*.PDF). To access files in this format you will need a copy of the Acrobat Reader. This software is available free of charge from the Acrobat site: <<http://www.adobe.com/>>.

The FirstClass area for MSC211 also includes regular announcements and copies of handouts. You should visit the site at regular intervals.

Resources you can purchase

On-campus students are advised to purchase:

Systems Implementation A: Study Guide

Systems Implementation A: Readings

Systems Implementation A: Visual Basic Summary

Off-campus students

Provided by Deakin University

- Study Guide
- Readings
- Suggested answers
- Visual Basic Summary
- Weekly tutorial exercises, which you will access via FirstClass conferencing

The FirstClass area for MSC211 also includes regular announcements and copies of handouts. You must visit the site at regular intervals.

Recommended textbook for all students

Bradley, J. & Millspaugh, A. 1999, *Programming in Visual Basic* Version 6, Irwin, Boston.

(1999 edition is suitable for MSC211 but for students planning on studying MSC338 the 2002 edition is recommended.)

The textbook is recommended rather than prescribed. A textbook such as that recommended for this unit is able to provide much more information about the features of the programming language than is possible in the study material for the unit. It will also provide examples of the techniques being used. If the recommended textbook is not available, any textbook on Visual Basic should suffice. The references in the notes will, of course, apply to the recommended textbook and if you are using another textbook you will need to consult its directory of contents to align its material to that being covered in the unit at any time.

Further resources

Barfield, L. 1993, *The User Interface, Concepts and Design*, Reading, Addison-Wesley.

Juliff, P. 1990, *Program Design*, 3rd edn, Prentice-Hall, Sydney.

Robertson, L.A. 1993, *Simple Program Design*, 2nd edn, Nelson, Vic.

Software

The unit requires students to have access to Visual Basic 6.0. On-campus students will have access to the software via the Deakin University Laboratories.

Off-campus students should acquire a personal copy of the software. Visual Basic Version 6.0 software is available as either the Working Model Edition, the Learning Edition or the Professional Edition. Note that the recommended textbook includes the Working Model Edition of Visual Basic 6. The Working Model Edition, is a cut down version of Visual Basic which does not allow you to create executable programs(.exes). This means the programs you develop in this unit can not be run without the Visual Basic program. Another limitation of the Working Model is that it does not include any Help files, however it can be used for developing the tutorial and assignment work in this unit, and it has the advantage of being supplied with the suggested text book. If you prefer you may purchase either the Learning Edition or the Professional Edition of Visual Basic. These products are available at academic prices, from the bookshop, and do not have the limitations of the Working Model. **Please note, we cannot accept assignment work in any other version of Visual Basic.**

Computer Mediated Communication—FirstClass

All off-campus students are required to use the computer conference service FirstClass. Your weekly tutorial work, demonstration program files, assignment details and examples will be made available to you through the FirstClass system. There will also be an opportunity to work with other off-campus students to complete review questions and to share ideas and solutions to practical problems. You can also use the conference service to send assessable tutorial work and assignments to your off-campus coordinator.

All off-campus students should have received the Deakin Learning Toolkit of software and documentation. This package includes 'FirstClass' the electronic conferencing software package. Your study materials package will also include a booklet called *Learning in the On-line Environment* that contains information on connecting and using the FirstClass system. If you have not received your package or have questions regarding the use of FirstClass please contact Christine Goodwin at the Warrnambool campus, (03) 5563 3149.

When you log on to FirstClass and enter the Bachelor of Commerce conference, you will find a number of conference areas. Open the LEVEL 2 UNITS folder, then the MSC211 folder. This is the main teaching area for this unit and contains some special purpose folders. One of them is for you to leave a 'sign on' message for this unit. When you sign on you will be allocated to a tutorial group. After you have been allocated to a tutorial group you will see another conference area labelled 'MSC211 Group n', where n is the number of your tutorial group. You will not be allocated to a tutorial group until you sign on.

If, after reading the documentation on setting up FirstClass, you have technical problems please refer these to the ITS help-desk. Your off-campus coordinator CANNOT help you with technical problems associated with setting up FirstClass.

Assignment details

You will be provided with assignment details via FirstClass. As mentioned earlier, each week you will be required to complete a Visual Basic tutorial. This practical work will teach you the skills you need to complete your assignments. After completing the weekly tutorial, we suggest you apply these newly acquired skills to your assignment. The assignments are of such complexity that it would be impossible to complete them unless you adopt this approach.

Screen resolution for Visual Basic assignments

All programs developed in Visual Basic are to be developed for a 1024 × 768 screen resolution. You may need to change the resolution on your home computer if you are using it to ensure you comply with this request.

Submission and assessment policy

On-campus students

- Completed assignments will comprise of a number of Visual Basic files and Word documents. These must be submitted to the office of the Faculty of Business and Law on your campus, on or before the due date.

Off-campus students

- Completed assignments will comprise a number of Visual Basic files and Word documents. These must be submitted electronically through FirstClass on or before the due date. Instructions on exactly how to submit assignment files will be posted to the FirstClass conferencing system (see section 'How to submit your assignment electronically').

All students

- Late submissions of assignments will not receive unit credit unless prior arrangement has been made with the MSC211 lecturer or unit coordinator. Assignments handed in late that have not been approved, will be marked but these marks will only be used in the final assessment, if you need them to pass the unit. Please note that once we have finished marking assignments and returned them to students we can not accept any more late submissions for that assignment.
- Fail grade submissions may be re-submitted. Re-submissions are encouraged. Marking of re-submissions will be handled in the same manner as that outlined for late submissions.
- It is the responsibility of the student to keep diskette backups and hardcopies of all work developed.

Assignment collaboration

In professional practice, software development is a collaborative activity. You will derive support and education from your colleagues. However, there is a distinction, both ethical and legal, between accepting help from another by asking for their advice and looking at their work as compared with simply copying their work with or without their permission.

Copying of software is plagiarism (see below) just as is copying of literature. This means that anything copied from another must have its source acknowledged. If you merely copy the work of another student, your assignment will not be marked. In addition, the student from whom you copied will not have their assignment marked either. This means that you must not allow your work to be copied by another student either by giving permission or by allowing your work to be obtained by another student through your carelessness in leaving around disks or listings.

By all means obtain help from tutors and friends but you must write the applications yourself and you must understand them in order to be able to pass the examination.

Plagiarism

Submitting written work, in whole or in part, that is copied or paraphrased from other authors (including students), without correct acknowledgment, is considered to be one of the most serious of academic crimes. This practice is equivalent to cheating in

examinations, and it may lead to expulsion from the University. For further information, you are referred to 'Regulations *Part 4–Academic Offences*, in the Deakin University handbooks, or Regulation 4.1(1), in 'University Organisation and Governance', <<http://dugs.deakin.edu.au/>>. Select the *Organisation and Governance* icon; (if you are prompted for a login, User ID is: *guest*, Password is: *guest*); then select Legislation; Legislation Index; open Regulation 4.1(1) *Student Discipline*; see Part 2–Academic Misconduct.

Please note that these regulations are not intended to discourage group work and exchange of views and information with other students and staff. Such interaction is most desirable provided you ultimately write your own answers and acknowledge any quoted sources.

How to submit your assignment electronically

Ensure that your assignment document has your name and student number at the beginning.

The process for submitting your assignments electronically is:

- 1 Click on the MSC211 Assignment Box to enter it. (Note you cannot read any messages in this area. You can post messages but not read them).
- 2 FirstClass software users—from the Message menu, choose New Message. Web browser users—click the New button in the conference window toolbar. The message will be automatically addressed to MSC211 Assignment Box.
- 3 In the Subject: field of the message header, enter Assignment number (e.g. 1, 2, etc.).
- 4 Copy and paste your prepared assignment into the message window, or attach the prepared file to the message.
FirstClass software users—use the Attach File command from the File menu.
Web browser users—click the Browse button next to the Attachments field of the message header.
- 5 Check that your message is addressed correctly (i.e. the To: field contains the correct unit code MSC211 Assignment Box). Send the message. An attached file is sent along with the message.

This process will send the message to the MSC211 Assignment Box. You all have the ability to send a message to this area, and you can open it and see the list of messages that are in it. But you can NOT read messages in the Assignment Box.

For your reference, a copy of your message is stored in your Mailbox on your FirstClass desktop. Your FirstClass Mailbox icon is on the entry level of the conference system. Messages remain in this Mailbox for 30 days, after which they are automatically deleted.

Assessment tasks

You will be assessed on both your practical skills in developing software and your knowledge of the practices and procedures involved in the software development process. For this reason, it is important that you pay attention to the theoretical content of the material in this unit as well as the practical skill in completing the assignments. Both of these aspects will be examined in the unit assessment.

Your assessment in this unit will be based on your performance in:

Assignment 1	Due date: 29 April 2002	15%
Assignment 2	Due date: 28 May 2002	15%
Tutorial tasks		10%
Final examination	Period commencing 17 June 2002	60%

The final examination is a closed book three-hour written examination paper.

Awarding marks and grades in this unit—the excellence rule

There are two assessable areas in this unit—continuous assessment (the assignments and tutorial work) and examination assessment.

In order to obtain a pass grade (P) in this unit, satisfactory work in both of the assessable areas must be achieved, and an *Awarded_Mark* of 50% or greater must also be achieved.

To obtain a higher grade (C, D, or HD), work of a high standard overall in the two sections is required. Note particularly that the unit team is not prepared to award a high grade when a low mark is achieved in the examination.

The unit team regards a mark less than 40%, in any of the two assessable areas (continuous assessment and examination), as unsatisfactory and irrespective of performance in other areas, a result of less than 40% of the marks allocated for any one of the assessable areas will result in a fail grade for the unit.

Details of how to calculate the *Awarded_Mark* follow.

A mark for continuous assessment, the *Continuous_Mark* will be calculated by the sum of the two assignments and the tutorial work (i.e. a total out of a possible 40).

The *Examination_Mark* is the actual mark obtained in the examination scaled to the mark out of a possible 60 (reflecting the weighting of exams in this unit relative to continuous assessment). You must get more than 40% in the examination in order to pass.

Next, a *Raw_Mark* is calculated as the sum of *Continuous_Mark* and *Examination_Mark*. The final mark allocated to students is the *Awarded_Mark*, and will be calculated as follows:

- (a) If unsatisfactory work is submitted in any of the assessable areas, that is ...

If your *Continuous_Mark* is less than 16 or

If your *Examination_Mark* is less than 24

Then the *Awarded_Mark* will be the smaller of 49 and the *Raw_Mark*

(b) Otherwise, if (a) does not apply, the *Awarded_Mark* will be the *Raw_Mark*

A grade, one of N, P, C, D or HD will be awarded according to the Deakin University scale based on the *Awarded_Mark* in the ranges:

Awarded_Mark < 50	N
Awarded_Mark >= 50 and Awarded_Mark < 60	P
Awarded_Mark >= 60 and Awarded_Mark < 70	C
Awarded_Mark >= 70 and Awarded_Mark < 80	D
Awarded_Mark >= 80	HD

Assignment 1

Due date: 29 April 2002

Marks: 15%

Assignment 1 details will be placed on FirstClass as soon as they are available.

Assignment 2

Due date: 28 May 2002

Marks: 15%

Assignment 2 details will be placed on FirstClass as soon as they are available.

Unit planner

As already mentioned, MSC211 is divided into 12 topics. It is recommended that you study one topic per week. These topics and the due dates for your assignments are given below. The unit planner indicates the week in which each topic will be covered in lectures. The deadlines set for your assignments must be met (see the 'Submission and assessment policy'). The date set for your final exam will not be negotiable.

Week beginning	Topic
4 March	1 Introduction to software development
11 March	2 Data definition
	Start working on Assignment 1 after completing tutorial 2
18 March	3 Simple procedure design
25 March	4 Introduction to sub-procedures
29 March–7 April	Intra-semester break
8 April	5 Program testing
15 April	6 Design of event driven software

22 April	7 Introduction to array processing
29 April	Due date: Assignment 1 Start working on Assignment 2 after completing tutorial 7
29 April	8 File organisation and processing
6 May	9 Searching and sorting arrays
13 May	10 Elegance in procedure design
20 May	11 Program documentation
27 May	12 Software quality
28 May	Due date: Assignment 2
3 June	Revision
8 June	Study period begins
17 June	Examination period begins
19 July	Assessment results released on WWW (StudentConnect) and telephone (IVR) from 6.30 p.m. in accordance with result release schedule
22 July	Results mailed to students