MACQUARIE
University
SYDNEY·AUSTRALIA

COMP229

Object-Oriented Programming Practices

S2 Day 2019

Dept of Computing

Contents

Disclaimer

Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.

General Information

Unit convenor and teaching staff
Unit Convenor
Dominic Verity
dominic.verity@mq.edu.au
Contact via contact via email
4RPD266
by appointment, except during recess

Lecturer
Matthew Roberts
matthew.roberts@mq.edu.au
Contact via contact via email
by appointment, except during recess

Tutor
Cameron Pappas
cameron.pappas@mq.edu.au
Contact via contact via email

Tutor
Damian Jurd
damian.jurd@mq.edu.au
Contact via contact via email

Tutor
Jarred Reilly
jarred.reilly@mq.edu.au
Contact via contact via email

Credit points
3

Prerequisites
COMP125(P)

Corequisites

Co-badged status

Unit description
Object-oriented programming is a key technology for modern computing. This unit bridges the gap between introductory programming and larger multi-person projects by considering the use of object-oriented techniques to produce intermediate sized software. Practical exercises emphasise the importance of programming practices such as appropriate documentation, systematic approaches to debugging and testing, and the use of software development tools. The unit is taught using Java.

Important Academic Dates

Information about important academic dates including deadlines for withdrawing from units are available at https://www.mq.edu.au/study/calendar-of-dates

Learning Outcomes

On successful completion of this unit, you will be able to:

Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

Have an ability to use application libraries, in this case the Java SE API

Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)

Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)

Apply key OO concepts and libraries to design and develop graphical applications

Apply key OO concepts of concurrency in working code.

General Assessment Information

LATE PENALTY

No extensions will be granted without an approved application for Special Consideration. There will be a deduction of 20% of the total available marks made from the total awarded mark for each 24 hour period or part thereof that the submission is late. For example, 25 hours late in submission for an assignment worth 40 marks – 40% penalty – deducted from the total for an assignment marked out of 100. No submission will be accepted after solutions have been posted.

EXAMINATIONS

The unit has three module examinations which assess the material studied in weeks 1-4, 5-9, and 10-12, respectively. Each examination is offered twice: once during the workshop classes in weeks 5, 10 and 13, respectively, and once in the final examination period. The repeat offerings of the examination will not be identical examinations but will be designed to assess the same material. All examinations delivered electronically via e-Learn in the computing laboratories on level 1 of building 9WW.

If a student attempts an examination more than once then the higher of their marks for the two attempts will be used to compute the grade.

If you receive Special Consideration for the final exam, a supplementary exam will be scheduled after the normal exam period, following the release of marks. By making a special consideration application for the final exam you are declaring yourself available for a resit during the supplementary examination period and will not be eligible for a second special consideration approval based on pre-existing commitments. Please ensure you are familiar with the policy prior to submitting an application. Approved applicants will receive an individual notification one week prior to the exam with the exact date and time of their supplementary examination.

Assessment Tasks

Name	Weighting	Hurdle	Due
In-class Exercises	16%	No	Weeks 1-4, 6-9, 11, 12
Module Examinations	44%	No	Weeks 5, 10, 13, Exam Period
Assignment One	8%	No	Week 4
Assignment Two	14%	No	Week 8
Assignment Three	18%	No	Week 12

In-class Exercises

Due: **Weeks 1-4, 6-9, 11, 12**
Weighting: **16%**

Each week you will be set a programming task to complete during your workshop class. You will be offered four (4) randomly selected opportunities for your in-class work to be assessed and your mark for this component will be computed using the best three (3) of those attempts. Your work will be checked and marked in workshop class in which it is completed. No late submissions are accepted.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging,

documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)
- Apply key OO concepts and libraries to design and develop graphical applications
- Apply key OO concepts of concurrency in working code.

Module Examinations

Due: **Weeks 5, 10, 13, Exam Period**
Weighting: **44%**

On three occasions during semester and once in the final exam period students sit a module examination. The three in-semester exams are run in practical workshops and each offers a different exam. In week 5 students will sit the Java Programming Practice exam, in week 10 students will sit the Design Patterns exam, and in week 13 students will sit the Concurrency exam. All three exams are then offered a second time in the exam period (with different questions of course). All exams are held in the computing laboratories and are completed online.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)
- Apply key OO concepts and libraries to design and develop graphical applications
- Apply key OO concepts of concurrency in working code.

Assignment One

Due: **Week 4**
Weighting: **8%**

The assignments are programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises or the early diagnostic test. 20% late penatly per day or part-day. Submissions are via iLearn.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)

Assignment Two

Due: **Week 8**
Weighting: **14%**

The assignments are programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises or the early diagnostic test. 20% late penatly per day or part-day. Submissions are via iLearn.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)
- Apply key OO concepts and libraries to design and develop graphical applications

Assignment Three

Due: **Week 12**
Weighting: **18%**

The assignments are programming exercises that allow skills to be demonstrated by solving a more substantial problem than in the weekly exercises or the early diagnostic test. 20% late penatly per day or part-day. Submissions are via iLearn.

On successful completion you will be able to:

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Apply key OO concepts and libraries to design and develop graphical applications
- Apply key OO concepts of concurrency in working code.

Delivery and Resources

CLASSES

Each week of COMP229 has one hour of lectures and a two-hour mixed class (i.e., two hour combined tutorial and practical). The Mixed classes commence in Week 1 and are held in the Computer Laboratory in room 9WW 121 (Wednesdays and Thursdays) or 9WW 123 (Fridays). In all cases students are expected to do significant preparatory work, readings and exercises *before* attending classes.

REQUIRED AND RECOMMENDED TEXTS AND/OR MATERIALS

The required text for the unit is:

- Head First Design Patterns by Eric T Freeman, Elisabeth Robson, Bert Bates and Kathy Sierra, O'Rielly Media, October 2004 (ISBN:978-0-596-00712-6)

Note that this text book was introduced for the 2011 offering of COMP229, so you may be able to purchase a second hand copy from a former COMP229 student. Failing that, you will be able to purchase a copy from the University COOP Bookshop or from your favourite online purveyor of fine literary works.

There will be no lecture notes provided, all examinable material is given in course readings and the textbook. Students are required to study this material and answer preparatory questions *before* class.

The Macquarie library contains many books on object-oriented programming in general, and on programming specifically in Java, that you may want to use to supplement the text and lecture notes.

One particularly useful service that the library provides you with is access to many Java related titles online via the Safari Books Online (http://proquest.safaribooksonline.com/) service. Using this service, which you can only access from a machine connected to the University network, you might like to have a look at the following Java titles:

1. Learning Java, 3rd Edition by Jonathan Knudsen; Patrick Niemeyer, ISBN: 978-0-596-00873-4
2. Java in a Nutshell, 5th Edition by David Flanagan, ISBN: 978-0-596-00773-7
3. Java Examples in a Nutshell, 3rd Edition by David Flanagan, ISBN: 978-0-596-00620-4

The web itself is an ideal source of Java information, and from time to time we will be posting useful links on the COMP229 iLearn site. Two particularly useful resources are:

4. Thinking in Java by Bruce Eckel, a free version of the 3rd edition of this pretty comprehensive book is available for download from http://www.mindview.net/Books/TIJ/ and its 4th edition, which is updated for use with Java 5 and 6, is available for $25 from http://mindview.net/Books/TIJ4.
5. The official Java Tutorial http://download.oracle.com/javase/tutorial/ which is a comprehensive resource providing trails covering topics ranging from the basics of Java programming to more advanced subjects like GUI development, Generics, Class Reflection, Sound, Graphics, Network Programming and Concurrency

UNIT WEBPAGE AND TECHNOLOGY USED AND REQUIRED

Online Resources

COMP229 will make extensive use of the iLearn system for delivery of class materials, discussion boards, real time chat, submission of work and access to marks and feedback. Students should check the iLearn site (http://ilearn.mq.edu.au) regularly for unit updates.

Questions and general queries regarding the content of this unit, its tutorials or practicals should be posted to the appropriate discussion board on the COMP229 iLearn site. In particular, any questions which are of interest to all students in this unit should be posted to one of these discussion boards, so that everyone can benefit from the answers.

Echo360

Audio and screen video recordings of the lectures will be made available online via Echo360. A link to these recordings will be provided on the iLearn site for this unit

Technology Used and Required

The practical work in this unit involves programming in Java (https://www.java.com) using the Microsoft Visual Studio Code IDE (https://code.visualstudio.com). We will also be using a distributed version control system called Git to access shared code repositories hosted on the BitBucket website (https://bitbucket.org).

This software is already installed for you in the 200 level computing labs (ground floor, 9 Wally's Walk) and it is available to download, install and use for free on your own machine(s). It should work equally well on Mac OSX, Linux or Microsoft Windows platforms.

Tools and libraries to support debugging, automated testing, GUI development and so forth will be introduced and used as the semester progresses. When that occurs you will be provided with full instructions in lectures on how to install and use each one.

Unit Schedule

week	lecture	lecturer	workshop	submit
1	Version Control	MR	Version Control	
2	Java Classes and Objects	MR	Java Classes and Objects	
3	Inheritance and Overloading	MR	Inheritance and Overloading	
4	Generics	MR	Generics	assign 1
5	Exceptions, Introduction to Patterns	MR	Exam: Java Concepts	
6	Strategy Pattern, Observer Pattern	MR	Observer Pattern	
7	Decorator Pattern, Iterator/Composite Patterns	DV	Decorator Pattern, Iterator/Composite Patterns	
8	Singleton Pattern, Command Pattern	DV	Singleton Pattern, Command Pattern	assign 2
9	Adapter and Facade Patterns	DV	Adapter and Facade Patterns	
10	Concurrency	DV	Exam: Design Patterns	
11	Concurrency	DV	Concurrency	
12	Concurrency	DV	Concurrency	assign 3
13	Review	DV	Exam: Concurrency	

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central). Students should be aware of the following policies in particular with regard to Learning and Teaching:

- Academic Appeals Policy
- Academic Integrity Policy
- Academic Progression Policy
- Assessment Policy
- Fitness to Practice Procedure
- Grade Appeal Policy
- Complaint Management Procedure for Students and Members of the Public
- Special Consideration Policy *(**Note:** The Special Consideration Policy is effective from 4 December 2017 and replaces the Disruption to Studies Policy.)*

Undergraduate students seeking more policy resources can visit the Student Policy Gateway (htt ps://students.mq.edu.au/support/study/student-policy-gateway). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

If you would like to see all the policies relevant to Learning and Teaching visit Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central).

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/study/getting-started/student-conduct

Results

Results published on platform other than eStudent, (eg. iLearn, Coursera etc.) or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in eStudent. For more information visit ask.mq.edu.au or if you are a Global MBA student contact globalmba.support@mq.edu.au

Special Consideration for Final Examinations

If you receive Special Consideration for the final exam, a supplementary exam will be scheduled after the normal exam period, following the release of marks. By making a special consideration application for the final exam you are declaring yourself available for a resit during the supplementary examination period and will not be eligible for a second special consideration approval based on pre-existing commitments. Please ensure you are familiar with the policy prior to submitting an application. Approved applicants will receive an individual notification one week prior to the exam with the exact date and time of their supplementary examination.

Plagiarism and Academic Honesty

Please refer to the Macquarie University academic integrity policy and the student code of conduct to find out more about what is expected from you with regard to issues such as plagiarism, deception, fabrication and sabotage. You should note that penalties for breaches of these policies, such as cheating in assignments or examinations, can be very severe; they range from loss of marks to failure in a unit (or units) and suspension from from the University. Please consult the student code of conduct FAQ for more information about what academic misconduct is, how allegations of misconduct are handled, how discipline committees work, and the kinds of panalties that can apply if you are found responsible for an act of academic misconduct.

You should be aware that we use a range of powerful source code analysis tools to detect cases of plagiarism amongst assignment and weekly exercise submissions. So if you plagiarise the work of others it is likely that we will catch you out and apply the strongest penalty that applies under the University’s academic honesty policy.

Student Support

Macquarie University provides a range of support services for students. For details, visit http://students.mq.edu.au/support/

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to improve your marks and take control of your study.

- Workshops
- StudyWise
- Academic Integrity Module for Students
- Ask a Learning Adviser

Student Services and Support

Students with a disability are encouraged to contact the Disability Service who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

If you are a Global MBA student contact globalmba.support@mq.edu.au

IT Help

For help with University computer systems and technology, visit http://www.mq.edu.au/about_us/offices_and_units/information_technology/help/.

When using the University's IT, you must adhere to the Acceptable Use of IT Resources Policy. The policy applies to all who connect to the MQ network including students.

Graduate Capabilities

Creative and Innovative

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

Assessment task

- Assignment Three

Discipline Specific Knowledge and Skills

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

Learning outcomes

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language
- Have an ability to use application libraries, in this case the Java SE API
- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)
- Apply key OO concepts and libraries to design and develop graphical applications

Assessment tasks

- In-class Exercises
- Module Examinations
- Assignment One
- Assignment Two
- Assignment Three

Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

Learning outcomes

- Be familiar with and apply good programming practices such as testing, debugging, documentation, version control, programming tools and interactive development environments (IDE)
- Understand the concepts underlying design patterns and apply a working knowledge of a selection of well known design patterns (singleton, observer, factory, iterator, facade, decorator etc)

Assessment tasks

- In-class Exercises
- Module Examinations
- Assignment One
- Assignment Two

Problem Solving and Research Capability

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

Learning outcome

- Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

Assessment tasks

- In-class Exercises
- Module Examinations
- Assignment One
- Assignment Two
- Assignment Three

Engaged and Ethical Local and Global citizens

As local citizens our graduates will be aware of indigenous perspectives and of the nation's historical context. They will be engaged with the challenges of contemporary society and with knowledge and ideas. We want our graduates to have respect for diversity, to be open-minded, sensitive to others and inclusive, and to be open to other cultures and perspectives: they should have a level of cultural literacy. Our graduates should be aware of disadvantage and social justice, and be willing to participate to help create a wiser and better society.

This graduate capability is supported by:

Learning outcome

- Apply key OO concepts of concurrency in working code.

Assessment tasks

- In-class Exercises

- Module Examinations
- Assignment Two

Changes from Previous Offering

2018 onwards: All module examinations moved to workshops. All examinations will now be held in the Computing Laboratories and will be delivered online via iLearn.

2019: Migrate to using Microsoft Visual Studio Code for Java development.

Assessment Standards

COMP229 will be graded according to the following general descriptions of the letter grades as specified by Macquarie University.

- **High Distinction (HD, 85-100)**: provides consistent evidence of deep and critical understanding in relation to the learning outcomes. There is substantial originality and insight in identifying, generating and communicating competing arguments, perspectives or problem solving approaches; critical evaluation of problems, their solutions and their implications; creativity in application.

- **Distinction (D, 75-84)**: provides evidence of integration and evaluation of critical ideas, principles and theories, distinctive insight and ability in applying relevant skills and concepts in relation to learning outcomes. There is demonstration of frequent originality in defining and analysing issues or problems and providing solutions; and the use of means of communication appropriate to the discipline and the audience.

- **Credit (Cr, 65-74)**: provides evidence of learning that goes beyond replication of content knowledge or skills relevant to the learning outcomes. There is demonstration of substantial understanding of fundamental concepts in the field of study and the ability to apply these concepts in a variety of contexts; plus communication of ideas fluently and clearly in terms of the conventions of the discipline.

- **Pass (P, 50-64)**: provides sufficient evidence of the achievement of learning outcomes. There is demonstration of understanding and application of fundamental concepts of the field of study; and communication of information and ideas adequately in terms of the conventions of the discipline. The learning attainment is considered satisfactory or adequate or competent or capable in relation to the specified outcomes.

- **Fail (F, 0-49)**: does not provide evidence of attainment of all learning outcomes. There is missing or partial or superficial or faulty understanding and application of the fundamental concepts in the field of study; and incomplete, confusing or lacking communication of ideas in ways that give little attention to the conventions of the discipline.

Please note that, since the beginning 2011, Macquarie University no longer awards Conceded Pass (PC) grades in its units.

The standards of achievement that will be used to assess each of the assessment tasks with respect to the letter grades are as follows. Where applicable, more specific versions of these standards will be provided with the assessment task descriptions.

Learning Outcome 1: Understand the key concepts of object oriented programming, and program proficiently in an OO programming language

P	Can correctly reproduce basic facts and definitions across a breadth of concepts and issues, but lacks depth of understanding.
Cr or D	Exhibits breadth and depth of understanding of concepts and issues. Can use terminology accurately in new contexts. Can express ideas in their own words and has an understanding of the limits of their understanding.
HD	As for Cr or D and is aware of the context in which the concepts and issues are developed and their limitations. Able to generate and justify principles and hypotheses for existing or new concepts or issues.

Learning Outcome 2: Have an ability to use application libraries, in this case the Java SE APII;

P	Can develop java programs that are similar to provided examples, and understands how to create basic tests.
Cr or D	Can create and understand designs that differ from provided examples and apply a variety of design techniques to their solution.
HD	As for Cr or D, and is proficient in the application of OO libraries, and application of error handling.

Learning Outcome 3: Be familiar with good programming practices such as testing, debugging, documentation and version control;

P	Can apply some basic documentation, debugging and testing practices along the lines of examples provided.
Cr or D	Is able to apply a wide range of documentation, debugging and testing practices to their code along the lines of examples provided. Has the ability to create non-trivial automated tests.
HD	As for Cr or D and has well-developed skills for applying documentation, debugging and testing practices in ways that have not been previously illustrated by examples.

Learning Outcome 5: Ability to apply key OO concepts and libraries to develop graphical applications;

P	Can put together code samples to create a non-trivial application involving a graphical interface, with some implementation of known patterns.
Cr or D	As for P and has a sound understanding of a number graphical concepts, and is able to apply them to new designs, with the help of additional material which has bot been discussed.
HD	As for Cr or D and can develop applications using techniques or approaches that have not been discussed.

Learning Outcome 4: Understanding of the concepts underlying design patterns and a working knowledge of a number of well known design patterns (singleton, observer, factory, iterator, decorator etc);

P	Can correctly reproduce basic facts and definitions relating to design patterns and can identify opportunities for their application, but lacks depth of understanding.
Cr or D	Exhibits breadth and depth of understanding of design pattern concepts and issues. Can use related terminology accurately in new contexts. Can construct implementations of common design patterns and apply them when writing code. Can express these ideas in their own words and has an understanding of the limits of their understanding.
HD	As for Cr or D and is aware of the context in which design pattern concepts are developed and their limitations. Able to generate and justify principles and hypotheses for existing or new design pattern concepts. Applies appropriate patterns consistently throughout the design and implementation of a complete software system, and can give a complete account of how those patterns interact to achieve a functionality objective.

Learning Outcome 6: Ability to apply key OO concepts and libraries to concurrent applications;

P	Can put together code samples to create a non-trivial application involving concurrency.
Cr or D	As for P and has a sound understanding of concurrency, and is able to apply it to new designs, with the help of additional material which has bot been discussed.
HD	As for Cr or D and can develop applications using techniques or approaches that have not been discussed.

These assessment standards will be used to give a numeric mark to each assessment submission during marking. The mark will correspond to a letter grade for that task according to the University guidelines. The final mark for the unit will be calculated by combining the marks for all assessment tasks according to the percentage weightings shown in the assessment summary above.

Completing the unit satisfactorily requires you to achieve a P grade (50% or more) for your mark.

Assumed Knowledge

This unit requires:

- Experience writing programs in the Java programming language. In particular, we will assume that you are familar with all of the basic features of Java, including the use of strings, arrays, loops, control structures (such as if statements), classes and objects, methods, and recursion.
- Experience using an IDE such as Eclipse http://www.eclipse.org or IntelliJ https://www.jetbrains.com/idea to develop simple Java applications.

Changes since First Published

Date	Description
23/07/2019	Corrected email address of the tutor Jarred Reilly, which was incorrect in the system when this unit guide was originally published.