Module Guide

1 Welcome

Welcome to M257 *Putting Java to work*. We hope that you will find the module interesting and rewarding.

Before you start to study the M257 units, you should carry out the following tasks.

- Use the contents checklist in your module mailing to make sure you have received everything you need for the start of M257.
- Read through this *Module Guide*. It will provide you with the information you need at the start of the module and gives you an overview of the module's content and assessment.
- Read Sections 1 to 3 of the *NetBeans Guide* and carry out Activities 1, 2, 3, 4 and 6 in the *Netbeans Guide*. If you are particularly interested in the NetBeans IDE and have the time, then please feel free to read more of this guide.
- Familiarise yourself with the *study planner* on the home page of the M257 website. Each week you will need to access the *study planner* and follow the outline contained in the weekly guide to see what reading, activities and e-learning content you will need to cover and what assessment is due. A downloadable study calendar is also available for you to print and refer to.
- Find out the date of your first tutorial (see Section 4 of this guide for further information).

Once you've completed these tasks locate Week 1 in the *study planner* and enjoy the module!

1.1 Structure of M257

The module materials include ten printed units of study material and two optional web-based units containing case studies.

Part of your study of the module will require you to undertake assessment. This assessment involves the following components:

- five interactive computer-marked assignments (iCMAs);
- three tutor-marked assignments (TMAs);
- an end-of-module examination.

The M257 *study planner* provides guidance on when you should be studying a particular component of the module and carrying out tasks such as completing an assignment and the assignment cut-off dates (i.e. the date by which you should complete and submit your work). A downloadable study calendar is also available for you to print and refer to.

1.2 M257 materials

The components of this module are as follows. Items marked with an asterisk are available *only* from the M257 website. All other items will be provided as printed text and/or on CD-ROM and will be sent in the module mailing, due to be despatched before the module start date.

- This *Module Guide**.
- The M257 website*, which is an integral part of the module and includes some module materials that are unavailable elsewhere.
- The *NetBeans Guide*, which tells you how to use the NetBeans software.
- The online *study planner** and printable *study calendar**
- Twelve units of study texts, incorporating study instructions, self-assessment questions (SAQs) with solutions, and activities with solutions. *Unit 11* and *Unit 12* are optional and web-based. The study text is the main teaching medium used for the module. Each unit will take, in general, around one to two weeks to study. The activities are available on the M257 website. PDF versions of *Units 1–10* are also provided on the M257 website.
- M257 software: the NetBeans Integrated Development Environment (IDE) and additional software to run emulations of Java programs on mobile devices.
- The M257 *Diagnostic quiz**, available through the online *study planner*.
- The *Glossary**, which provides a description of all the technical terms used in the module.
- The *Exam Handbook*, which contains some core information about Java. The handbook may be taken into the examination. No annotations of the *Exam Handbook* are permitted apart from published errata.
- The *Index**, which provides references to the main topics in the module.
- iCMAs* (interactive computer-marked assignments): five computer-based tests that are completed online via the M257 website.
- TMAs* (tutor-marked assignments), which are assignments that you will have to complete as part of the module assessment. There are three TMAs on this module. Your tutor will mark these assignments and return them to you with some feedback to guide your study. The assignments are also used as a medium for teaching.
- A three-hour examination at the end of the module.
- The M257 website *News* page*: the *News* page enables the

module team to communicate directly with all students and alert you to any relevant information, errata or other module news. You should check the website for news items at least once a week.

2 Before you start

Please read the rest of this guide carefully to ensure that you are properly prepared for study. Remember that this module will take considerable time and effort on your part. We would normally recommend that an average student should allow at least six to seven hours per week to study M257.

Prerequisite knowledge

M257 is a level 2 module and as such we begin with the assumption that you already have some basic ideas of object-orientation. You might know about these from a previous module, from your work, or perhaps from something you have read independently.

You should be familiar with at least some of the following concepts:

- objects;
- objects interacting by invoking methods (exchanging messages);
- classes;
- inheritance;
- simple object-oriented design;
- control structures such as `if` statements.

You could have obtained this knowledge from studying OU modules such as M206, M263, M255 or other equivalent modules at another university.

Try our M257 *Diagnostic quiz* to see if you have the required knowledge. You will find a link to the M257 *Diagnostic quiz* on the M257 website home page. If you find the material tested here is unfamiliar to you then you should consider studying M255 (which covers all this material) before studying M257.

If you have any doubt about the level of study please seek advice from your Regional Centre.

Computing facilities

It is essential that you have the necessary computing equipment and specified software installed before you start the module. In particular, you need to ensure that you have:

- access to a computer that meets the minimum specification for the module (see below);
- obtained access to the internet by subscribing to an internet service provider (ISP);

- word-processing software for completing your assignments that produces output that can be read by Microsoft Word 97;
- installed the supplied software as explained in the *NetBeans Guide*.

You can find the computer specification required for M257 by going to the M257 *Putting Java to work* page of the Study at the OU website and clicking the 'Technical Requirements' link in the 'Study materials' section.

If you do not have a suitable version of Microsoft Word, you will still be able to read your marked assignments using the OpenOffice.org software provided on the 'Software downloads' pages of the OU Computing Guide, which you can access from your StudentHome page (see below).

Further information

You can find introductory information on general study skills through the Help section of your OU StudentHome page. You can use this resource as much or as little as you wish.

3 Overview of M257

Since its introduction, Java has become one of the most widely used computer languages. Initially riding on the crest of the rapid expansion of the internet, Java has established itself as a tremendously versatile programming language which is able to run on a wide variety of computer platforms.

Java can be used to build systems that power applications such as e-commerce, online games, downloadable music and online banking. Java runs on PCs, laptops and mobile devices (including phones, PDAs and tablets) and can be found in TV set-top boxes, in embedded devices (forming part of larger systems such as cars, robots or printers), in smart cards and even in wearable computers.

The module will show you how, through the concept of Java editions, Java can be used in all of the platforms mentioned above. Java editions cater for the different needs and resources of, say, large international e-commerce systems running on extensive networks with many servers, and games software running on mobile phones with very limited hardware resources. The module is mostly interested in the Java 2 Standard Edition (J2SE) but you will also be introduced to the Java 2 Micro Edition (J2ME), designed for use on smaller systems with limited resources (such as mobile phones).

The module will provide a broad coverage of the Java programming language from primitive data types and control structures, through graphical user interface (GUI) construction and event handling, to topics such as threaded programming and basic internet programming using sockets. Although the module will concentrate on producing stand-alone applications, it will also show you how the skills and knowledge that you develop are directly transferable to web-based systems through the use of applets and also to mobile phone systems through the use of MIDlets. The module will also touch on topics such as Java on servers (servlets) and even the idea of Java code roaming the internet through the use of aglets.

3.1 Aims and learning outcomes

Aims

The module aims to:

- reinforce the fundamental aspects of object-oriented technology;
- describe how these concepts are implemented in Java;
- provide the knowledge necessary to construct Java programs;
- describe a number of the advanced facilities of Java including exceptions and internet-based applications;
- show the implications of implementing Java applications on a range of platforms;
- show how Java can be used in developing non-trivial programs.

Learning outcomes

M257 provides opportunities for you to achieve the following general learning outcomes.

Knowledge and understanding

After studying the module, you will be able to demonstrate:

- an understanding of the design and programming processes;
- some knowledge of the main constructs and mechanisms in Java;
- an appreciation of the implications of implementing Java applications on a range of platforms;
- techniques used in developing a large Java program.

Cognitive skills

After studying the module, you will be able to:

- describe and apply key concepts and techniques in software design and development;
- analyse and abstract away from the details of a problem;
- design and formulate an appropriate solution to a problem and evaluate it.

Key skills

After studying the module, you will be able to:

- find information from a range of sources to support a task;
- plan complex tasks;
- use new Java libraries;
- use appropriate numerical, mathematical and abstraction skills;
- progress to more advanced-level studies.

Practical and/or professional skills

After studying the module, you will be able to:

- assemble, program, develop, test and evaluate software systems;
- use software tools such as a Java IDE;
- use appropriate programming skills;
- develop user interfaces appropriate for particular platforms.

3.2 Summary of M257 units

There are twelve module units and they are as follows:

Unit 1 Java everywhere

Unit 2 Java in the small

Unit 3 Java in the large

Unit 4 Input, output and exceptions

Unit 5 Packages and abstraction

Unit 6 Graphical user interfaces

Unit 7 Event-driven programming

Unit 8 Threads

Unit 9 Internet programming

Unit 10 'lets do Java

Unit 11 Web-based case studies

Unit 12 Web-based case study

An overview of each unit is given below. *Units 1*, *2* and *3* contain the background knowledge we assume you have when starting the module. They provide revision, and we would expect you to work quickly through these comparatively short units.

Unit 1 Java everywhere

This unit sets the main themes of the module; it describes the origins of Java as a web technology and the aims of the language designers. It also introduces a number of important points that we have to consider when looking at systems as a collection of interacting objects, invoking methods on each other. Objects in a system can also be composed of other objects.

The unit also introduces a deceptively simple idea: that Java contains a mechanism known as inheritance that enables a class to use methods and instance variables from other classes. This is an important mechanism for reuse in an object-oriented programming language like Java. We also describe an important graphical device known as an inheritance hierarchy, often called a class hierarchy. This shows how classes are related to each other in terms of the methods and instance variables they inherit from each other.

Unit 2 Java in the small

We have called this unit ‘Java in the small’ after a term coined by DeRemer and Kron in a paper presented in 1975 (DeRemer, F. and Kron, H. (1975) ‘Programming-in-the large versus programming-in-the-small’, *Proceedings of the international conference on Reliable software*, pp. 114–21). This refers to the programming of the internal parts of programs, in particular the contents of methods.

The previous unit introduced the idea of objects communicating through the invocation of methods on objects. In this unit you will learn:

- how to write the internal parts of methods;
- the different types of data that the language provides;
- the facilities for altering the flow of program execution.

Our objective is to create code that will be easy to read (either by the original writer or by somebody else) as well as easy to maintain. Of course, it must also be correct!

Unit 3 Java in the large

The previous unit described the ’small’ programming structures that languages such as Java offer. We have been concerned largely with how we store and operate on data, and how methods achieve their purposes. In this unit, we aim to show that Java is truly an object-oriented language. Object-oriented languages provide better support for what can be called ’programming in the large’, a very different skill to the kind of programming described in *Unit 2*.

The key object-oriented structure is the class, with its definitions of methods and variables. At the ’large’ level, we are also able to describe relationships between classes, such as inheritance.

Programming in the large is about specifying how the parts of our programs should work with one another and object-oriented languages provide us with keywords that allow us to reason about how methods may be used, how data may be manipulated and how objects may ’communicate’ with one another.

Another key topic in this unit is the issue of how to create objects from class definitions, specifying the starting state of the objects in detail.

Unit 4 Input, output and exceptions

Communication is vital. It is also potentially unreliable. This is true of human communication, but also of communication between humans and computers, or equally when one computer communicates with another across a network.

In this unit we look at communicating through files, a keyboard or another computer. Java provides a rich and flexible set of facilities for sequential input and output, within a consistent framework. This makes the process of communication similar, whatever the source or destination of the data.

Many things can go wrong – for example the expected file may be missing or the network links may be faulty. So, in this unit, we also take the opportunity to consider how to deal with error conditions and unexpected events.

We also discuss exception handling, the main approach in Java for dealing with serious or unpredictable error conditions. This can be applied to many different situations, but is particularly relevant to communication and input/output since unexpected events may readily occur.

Unit 5 Packages and abstraction

This unit considers a number of facilities of Java that are particularly useful for the structuring of larger systems.

The Java language provides the concept of a 'package' as a way to group a number of related classes. Large programs typically consist of a number of packages – this helps with management of the software development and maintenance processes.

All Java programs carry out a number of standard functions that are commonly required by many other programs. In recognition of this, Java contains a number of predefined packages, known as the standard Java packages. These form libraries of classes available for programmers to use. They provide a wide range of facilities including input/output, mathematical functions, graphical user interface components and data structures. A program normally has one or more packages of classes specially written for that application by the developer, together with classes from a number of standard Java packages.

This unit will also explain two further concepts – abstract classes and interfaces – which are helpful in structuring hierarchies of related classes. Abstract classes are useful in defining groups of classes related by inheritance. Interfaces are a powerful way of specifying what a class should do, without specifying in detail how it should implement this.

Unit 6 Graphical user interfaces

Previous units have described how data such as that from a file could be communicated to a Java program. However, there is also a need to handle data that is communicated in more diverse fashions: in particular, directly from a human. Such data is often taken into a system via a graphical user interface (GUI).

GUI design is a major field of study, and creating a ’usable’ interface is crucial to the development of a successful software system. The interfaces that you will create in this module will be relatively simple and will not require a great deal of planning and design. Typically, these interfaces will include buttons, areas for typing in text, drop-down menus for selecting options, areas for displaying results, and so on. These are known as visual components.

In this unit we will look at how such interfaces can be created and displayed. *Unit 7* will show how we can then get such interfaces to respond to the user, interacting with them through what is known as event-driven programming.

Unit 7 Event-driven programming

Unit 6 looks at how graphical user interfaces that contain a wide range of components and have quite sophisticated layouts are created. The creation of a link between something happening to the graphical user interface – an event – and the execution of code, commonly known as event-driven programming, is the subject of this unit. Event-driven programs are programs that respond to events initiated by the user, such as mouse clicks and key presses. This unit also introduces graphics.

Unit 8 Threads

It is often the case that you would like a program to be able to do more than one thing at a time. You might, for example, want to monitor the keyboard for a key being pressed by a user and, at the same time, track the movement of the mouse by the user and simultaneously repaint the screen. Each one of these tasks can be thought of as a single ’thread’ in a program. These threads all exist in the same executing environment and so are not the same as different programs. Each thread is a sequence of code that can be running simultaneously with others, each being between its start and finish at any point in time.

Java provides a great deal of support to the creation of such independent tasks. Each task is made into a thread, and on a multiprocessor computer each thread can run at the same time as all of the other threads (up to the number of processors being available, of course). However, on a single-processor computer, the CPU can carry out only one set of instructions at once. In this situation the threads share the CPU, with the operating system allocating small blocks of time to each thread. This gives the illusion to each thread that it has sole access to the CPU. As well as making programs more responsive, threads can also be very useful in making use of the times when a processor is idle because it is waiting for something to happen.

Unit 9 Internet programming

This unit describes how we use Java to program internet-based applications, such as those associated with e-commerce or internet chat rooms. Amongst other topics, we will look at how the internet is structured, the roles of clients and servers, and how they are programmed using sockets. We will also show how to make use of threads to enable servers to deal with multiple clients.

Unit 10 ’lets do Java

The underlying theme of this module is that Java is everywhere not only because it is a highly portable programming language, but also because it is able to run on a very wide range of platforms. In this unit we will be looking at how Java is able to work on such a wide range of platforms and environments. This unit looks at using Java in web-based systems, using applets. It looks at how Java can be used in small limited-resource systems, such as mobile phones, using MIDlets. We will also briefly consider server-side processing with servlets and ’mobile’ code using aglets, and meet the use of Java in embedded devices and smart cards.

Unit 11 Web-based case studies

This unit contains two case studies: a spider that roams the World Wide Web looking for information, and a Java interface to the Google search engine. These case studies show many of the facilities of Java in action and, in particular, show how Java is used in internet-based applications. This is an optional unit: you have the choice as to whether you wish to read it.

Unit 12 Web-based case study

This case study is a mobile application. It demonstrates the power of the J2ME framework. Like the previous unit it contains three activities for you to complete. This is an optional unit: you have the choice as to whether you wish to read it.

3.3 M257 software

The module has one CD-ROM – the NetBeans CD-ROM. The NetBeans CD-ROM contains the Java IDE, including an extension that you will need for *Units 10* and *12* enabling you to emulate Java programs running on a mobile phone. Installation instructions for the NetBeans software are provided on the case of the NetBeans CD-ROM.

At various points in the units, there will be practical activities for you to carry out in order to consolidate your understanding of the material. These will be indicated by a keyboard icon in the margin and the relevant activity number (an example is shown in the margin here). The activities for *Units 1–12* are available on the M257 website.

Activity 2.1

Experimenting with data types and operators. *(Example)*

Installation instructions for all the module activities (including the NetBeans activities) are given on the M257 website.

4 Study support

Tutorial support

As an M257 student you will be allocated to a tutor, who will help you with your study of the module throughout the year. Generally your tutor will live in the same geographical area as you do and will also be responsible for looking after a number of other students on the same module. Information about your tutor, such as his or her name, telephone number and email address will be available on your StudentHome page.

Your tutor is there to provide support and feedback in a number of ways, in particular by organizing tutorials and by marking and giving feedback on your assignments.

StudentHome is also the place to look for information about when any tutorials or day schools will be held.

Assignment marking

Correspondence tuition is at the heart of your study with The Open University. It is the main way in which you will be able to judge what progress you are making on the module. Doing the assignments will allow you to review the material in the relevant units and to consolidate your knowledge and skills.

For TMAs, your tutor has the responsibility for reading and assessing carefully what you have submitted. You will receive detailed individual feedback and guidance on your progress, as well as the overall mark that you have gained for the assignment. This is the most personalized aspect of your tuition and you should ensure that you make use of it. An important point to make about TMAs is that they provide your tutor with the opportunity to teach you as well as to assess your work. You should get your marked assignments back about two weeks after the cut-off date. Even though you will already be well into the study of the next unit, you should make time to look at your tutor’s comments rather than just look at the mark you have been given. There will usually be comments, suggestions and clarifications of misunderstandings that will be beneficial to your understanding of the module.

Tutorials

For most people there will also be the opportunity to meet their tutor and fellow students at tutorial sessions, which will be organized by your Regional Centre. They will be held at one of the OU tutorial venues and each session will last about a couple of hours. Details relating to all the tutorials for your module can be found on StudentHome. There may be occasions when you cannot attend the tutorials given by your allocated tutor, but you can attend similar tutorials given by other tutors from the same module at another centre. Generally this is possible, but it is always polite to contact the other tutor in advance. Although attendance at tutorials is optional, you are advised to attend if at all possible. They provide a valuable opportunity for you to meet other students, and to compare notes on the module materials and common problems of study that you are experiencing. You will also find that your tutor is able, in these sessions, to focus your attention on important issues for the current unit that you are studying, to answer questions you may have about its contents, and to clarify what is expected from you for the questions in the current assignment. Your tutor may also take the opportunity to go over the main issues and any misunderstandings that have been identified from your previous assignment. If there are areas of study that you wish to discuss in a tutorial, you may find it useful to suggest this to your tutor in advance. This will help tutors to plan their tutorial sessions; otherwise they may have to guess the problems that you are experiencing.

Day schools

Depending on the region in which you are based, you may have the opportunity to attend one or more full-day tutorial sessions, thus enabling you to take part in a more extended group study of the module materials.

M257 forums

You can also meet fellow students electronically. At the start of the module, you will be given access to the M257 forums. You will see links to the various forums on the M257 website home page. The forums can be used to discuss academic and other module-related issues with your fellow students and will be supported by one or more tutors. You will also have access to a forum run by your tutor for your tutor group. These forums are very important, and you should access them regularly.

Telephone and email tuition

In addition to face-to-face and correspondence tuition, you may need to contact your tutor from time to time about a specific problem that you have encountered. You should endeavour to use this mode of contact sparingly and save it for urgent queries. Generally, your tutor works for The Open University on only a part-time basis and does not have unlimited time to devote to the module. So when you contact your tutor from time to time, you need to ensure that you are using the time effectively. It may be easier for you to send an email message first and to follow it up later with a phone call when your tutor has had time to think about the problem that you are experiencing.

5 Assessment

Part of your study of the module will require you to undertake assessment. You should refer to the Assessment Handbook for information about the Open University’s policy on assessment.

You can access the handbook online from your StudentHome page.

There are two types of assignment used on M257 – iCMAs and TMAs.

> iCMAs: M257 has five iCMAs that are designed to help improve your understanding and knowledge of the concepts and principles covered in the first five units of M257. They are also intended to give you an early indication of how you are coping with the module. The iCMAs are completed online via the M257 website. Your iCMAs are marked automatically and your scores and feedback will normally be made available about a week after each iCMA cut-off date.
>
> TMAs: there are three TMAs on M257. TMA 01 is *formative*, meaning that although we highly recommend you complete and submit it to consolidate your knowledge of the first five units, the marks will not count towards your final score. TMAs 02 and 03 are *summative* and count towards your final score for this module.

All units except *Units 10*, *11* and *12* are assessed by the TMAs and iCMAs. *Units 1–5* are each assessed by an iCMA and by TMA 01. *Units 6–9* are assessed by TMA 02 and TMA 03. *Unit 10* is assessed in the examination.

Most TMA questions involve practical computing work.

There are three TMAs. The first TMA is *formative*, which means that the marks awarded will not count towards your final result. However, you are strongly encouraged to complete and submit the assignment in order to gain practical experience with the NetBeans IDE and to receive valuable feedback from your tutor, which you will be able to build upon in the rest of the module. If you have recently studied M255 and gained (or expect to gain) a pass 1 or 2, or if you have considerable experience with Java already, you will already have a good knowledge and understanding of the material covered in M257 Units 1 to 5, and you may therefore choose not to submit TMA 01..

TMA 02 and TMA 03 are both *summative*. The marks from TMAs 02 and 03 and the five iCMAs are combined with your examination score to determine your final module result. All continuous assessment components will be marked out of 100, *but they are not equally weighted in their contribution to your overall module score*. Table 5.1 shows the individual weightings for each continuous assessment item.

Table 5.1 Continuous assessment weightings

Component	Weighting (%)
iCMA 41	7%
iCMA 42	7%

iCMA 43	7%
iCMA 44	7%
iCMA 45	7%
TMA 01	0%
TMA 02	35%
TMA 03	30%

The iCMAs and TMAs must be submitted by the dates, known as cut-off dates, as shown on the study planner. iCMAs are completed and submitted online; TMAs are submitted to your tutor and marked electronically.

The end-of-module examination consists of a three-hour unseen paper, potentially covering the entire module, except *Units 11* and *12*.

The examination will consist of short-answer questions and a choice of long-answer questions. You will not be permitted to take any notes or units into the examination with you, except for the unannotated printed copy of the M257 *Exam Handbook* that was supplied to you.

In order to pass M257 you will need to achieve a minimum score of 40% in both the continuous assessment component of the module *and* the examination. The continuous assessment component and the examination each contribute 50% towards your final module grade. Details of how your final grade will be determined are given in the Assessment Handbook.

Note, however, that the system of substitution does not apply to this module.

You will be sent details of the location, time and date of the examination at a later date.

A specimen examination paper is available on the module website. This will aid you in understanding what is expected of you in the examination.

6 How to get help

There are many options available to you for obtaining help during your studies. For academic queries your first point of contact is your tutor, who will advise you at the start of the module how long it will usually take to respond to queries.

For module-related problems you should contact the M257 Curriculum Manager. You can contact the M257 Curriculum Manager via the faculty website at http://www.mct.open.ac.uk – click on the ‘Enquiry’ link.

For all other queries please see the Help section in StudentHome.

The OU Computing Helpdesk

The Open University offers a Computing Helpdesk service to all students currently studying a module with a computing element. The Helpdesk deals with more general module-related technical computing queries and can also help if you have problems using your OU computer username and password to gain access to any of the University's networked computing services.

If you have a computing problem the first place to start is by looking at the help available under the Help tab in StudentHome. If you are still experiencing problems, can contact the Computing Helpdesk via the Computing Helpdesk contact page.

Note. When contacting the helpdesk by email or telephone, please supply your Personal Identifier and module code (M257), together with the full and exact text of any error messages, etc. that your computer/software has given.

OU Library services

As a registered OU student, you are entitled to use all of the online library resources and services available to you via the Open University Library. These include databases of academic journals and journal abstracts, electronic books, newspapers, standards, statistics and industry/market reports.

You may wish to make use of Library resources in your study of M257 and you are encouraged to explore the Library website to see what is available to you. A full set of links to the Library's main site, the Library helpdesk, to information search guides and a guide to using the Library are contained in the 'Library resources' section of the M257 website, and through your StudentHome home page.

◀ Module Guide

Copyright © 2011 The Open University

Validate HTML Section 508 Check WCAG 1 (2,3) Check
You are logged in as Phil Molyneux | StudentHome