

Chapter 1

Introduction

Textbooks

There are many books on Java available on the market. You may have already had some books for the first year Java course cis109 and not yet finished reading them. We therefore do not recommend any specific text except the lecture notes for essential reading and a few books for desirable reading. A list of books for further reading and for supporting and historical background is also included below.

You do not need to read all the books on the lists but may choose one that suits you for frequent reference. To achieve the best efficiency to study the course unit cis212a, you need to follow the order and the reading instruction for each chapter in the lecture notes.

Reading

This is a summary of recommended readings for the course unit.

Essential reading

Pu, Ida *Lecture notes for cis212a Programming: Advanced Topics and Techniques (term 1)*, (9 2002)

Desirable reading

We recommend that you have ONE of the following books for frequent reference.

Bailey, Duane *Java Structures*. (WCB/McGraw-Hill, 1999) [ISBN 0-07-289179-3].

Bishop, Judy *Java Gently*. (Addison-Wesley Publishing Company, 2001) third edition [ISBN 0-201-71050-1].

Carrano, Frank M. and Prichard Janet J. *Data Abstraction and Problem Solving with Java*. (Addison-Wesley Longman 2001) [ISBN 0-201-70220-7].

Deitel & Deitel *Java - How to Program*. published by Prentice Hall, 1999) third edition [ISBN 0-13-287426-1].

Eckel, Bruce *Thinking in Java*. (Prentice Hall, 2000) second edition [ISBN 0-13-027363-5].

Goodrich, Michael T. and Tamass, Roberto *Data structures and algorithms in Java*. (New York; Chichester: Wiley, 2001) [ISBN??].

Main, Michael *Data Structures and Other Objects Using Java*. (Addison-Wesley Longman 1999) [ISBN 0-201-35744-5].

Rowe, Glenn W. *An Introduction to Data Structures and Algorithms with Java*. (Prentice Hall, 1998) [ISBN 0-201-35744-5].

Watt, David A. *Java collections :an introduction to abstract data types, data structures and algorithms*. (Wesly, 2001) [ISBN??].

Web addresses

Installing Java system

<http://burks.bton.ac.uk>
<http://java.sun.com/>
<http://textpad.com>

David J.Eck's Java course?
<http://math.hws.edu/eck/cs124>.
<http://math.hws.edu/javanotes/>

Java Systems
<http://www.javasoft.com>
<http://www.javasoft.com/products/hotspot/>

Java Gently
<http://www.cs.up.ac.za/javagently/>

Java for students
<http://www.cms.shu.ac.uk/java/>

Thinking in Java
<http://www.bruceeckel.com>

The Java Tutorial
<http://java.sun.com/docs/books/tutorial/>

A collection of Java readings
<http://www.maththinking.com/boat/javaBooksIndex.html>

Further reading

These books are also of interest and are recommended, but you will be fully prepared for the examination should you have studied only those texts listed above. These are provided for completeness and to allow the interested reader to pursue some of the topics in more depth. You will not be examined on the content of those books or papers listed in the references other than where the material appears in the texts listed above.

Barnes, David J *Object-oriented Programming with Java*. (Prentice Hall, 2000) [ISBN 0-13-086900-7].

Bell, Douglas and Parr, Mike *Java for Students*. (Prentice Hall, 1999) second edition [ISBN 0-201-31451-7].

Budd, Timothy *Understanding Object-oriented programming with Java*. (Addison Wesley Longman, Inc., 1998) first edition [ISBN 0-201-30881-9].

Kanerva, Jonni *The Java FAQ*. (Addison Wesley Longman, Inc., 1997) [ISBN 0-63456-2].

Koffman, Elliot and Wolz, Ursula *Problem Solving with Java*. (Addison Wesley Longman, Inc., 1999) [ISBN 0-201-35743-7].

Horstmann, Cay *Computing Concepts with Java 2 Essentials*. (John Wiley & Sons Ltd., 2000) second edition [ISBN 0-471-34609-8].

Laszlo, Michael J. *Object-oriented Programming featuring Graphical Applications in Java*. (Addison Wesley Longman, Inc., 2001) [ISBN 0-201-31451-7].

Lewis, John and Loftus, William *Java Software Solutions - Foundations of Program Design*. (Addison-Wesley Publishing Company, 2000) second edition [ISBN 0-201-61271-2].

McFarlane, Nigel and Chiarelli, Andrea etc. *Professional JavaScript*. (Wrox Press Ltd., 1999) [ISBN 1-861002-70-X].

Morelli, Ralph *Java, Java - Object-oriented Problem Solving*. (Prentice Hall, 2000) [ISBN 0-13-011332-8].

Savitch, Walter *Java - An Introduction to Computer Science & Programming*. (Prentice Hall, 1999) [ISBN 0-13-287426-1].

Shachelford, Russell L *Introduction to Computing and Algorithms*. (Addison Wesley Longman, Inc., 1998) [ISBN 0-201-31451-7].

Smith, Michael *Java - an Object-Oriented Language*. (McGraw Hill International (UK) Limited, 1999) [ISBN 007-709460-3].

Walrath, Kathy and Campione, Mary *The JFC Swing Tutorial*. (Addison-Wesley, 1999) [ISBN 0-201-43321-4].

Winder, Russel and Roberts, Graham *Developing Java Software*. (John Wiley & Sons Ltd., 1998) second edition [ISBN 0-471-60696-0].

Supporting and historical background

Bishop, Judy *Java Gently*. (Addison-Wesley Publishing Company, 1998) second edition [ISBN 0-201-342979].

Bishop, Judy *Java Gently*. (Addison-Wesley Publishing Company, 1997) first edition [ISBN 0-201-59399-8].

Brown, David *An Introduction to Object-oriented Analysis: Objects in Plain English*. (John Wiley & Sons, Inc., 1997) [ISBN 0-471-11028-0].

Budd, Timothy *Understanding Object-oriented programming with Java*. (Addison Wesley Longman, Inc., 1998) first edition [ISBN 0-201-30881-9].

Lecture notes

A collection of the lecture notes (Notes in short) covers the main topics in the syllabus. It can be used as a reference which summarises, highlights and draws attention to some important points of the subject. It is meant to cover the materials in the first term second-year BSc in CS/CIS/IC course at Goldsmiths which totals twenty hours of lectures and ten hours of laboratory sessions.

The Notes is not a course textbook. It essentially sets out a sequence in which to study the topics in the course within the limited study hours. Where coverage in the textbooks is weak it provides some additional background material including examples. It also provides guidance for further reading, particularly in those areas which are not covered adequately in the recommended textbooks.

It is unnecessary to get hold of every textbook recommended above. One or two books should be enough to enable individual topics to be studied in depth. One effective way to study programming in Java is to try by yourself and practise as much as possible. Exercises and coursework are good opportunities to help understanding. The sample examination paper at the end of the Notes may also provide useful information about the type of questions you might expect in an exam.

One thing the reader should always bear in mind is the fact that Java, like any other programming language, has kept evolving and has been update, sometimes at an annoyingly rapid pace. Therefore, some of the descriptive information provided in any texts about Java will eventually become outdated. You should, therefore, try not to be surprised if you find different approaches, explanations or results among the books you read including this Notes. Remember, you are not just learning how to write a program; you are learning how to write **good software**. This process requires the **input** of your own experiments and experience. Hence, you are encouraged to, if possible, peruse articles in trade journals, browse the relative Web sites, read the latest versions of books, attend conferences or trade shows etc., and in general pay attention to what is happening in the computing world.

At the end of each chapter, there are working sheets for labs.

About CIS212a

Our *aim* is to give students an insight into object-oriented programming design.

The *objectives* are:

- To introduce issues in programming such as Object-orientation, Recursion, Abstract Data Types (Data Structures) and Algorithms;
- To further develop the students' skills in Java programming;
- To introduce some advanced topics in the object-oriented platform such as graphical user interfaces and multi-threads.

Learning outcomes

On completion of the first term of this course, students should be able to:

- Understand the concepts introduced such as Object-orientation; Recursion, Abstract Data Types (Data Structures) and Algorithms;
- Learnt the techniques for identifying and solving a computing problem;
- Design and analyse Java programs at advanced level.

Prerequisites

The students should have already some prior experience with programming in Java, because cis212a builds on cis109 (Introduction to Computer Programming) developing the following aspects:

Inheritance, constructor methods, method overloading, method overriding, arrays and sorting, dynamic structures, types versus classes.

The students should also have an easy access to a Java platform or have a Java platform installed on their computers at home.

Installing Java

The best way to learn programming is to work on a computer. It is important to have access to a Java platform. You are strongly recommended to install a Java system on your home machine.

There are a lot of public domain versions of Java among which the most popular one called JDK (free) is at <http://www.javasoft.com/> or <http://java.sun.com/>. A great amount of information is provided on these sites and you can download the software.

If you are using Linux, then the free software package RedHat 6.0 or above already includes a free Java platform.

Testing the Installation

An easy way to test your installation is to type the following at a command prompt:

```
java -version
```

If you have JDK 1.2 installed, for example, a message in response should be seen on the screen containing at least the following:

```
java version "1.2"
```

This may be followed by a few more lines providing some further information.

For example:

```
java version "1.3.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.0)
Classic VM (build 1.3.0, J2RE 1.3.0 IBM build cx130-20000623
(JIT enabled: jitc))
```

Another example:

If you see a response (error) such as `bad command or file name` or `command not found` to your command `'java -version'`, you know that your installation has not been completely successful. This sometime may be simply because your system cannot find the correct version of the file which runs Java programs. You may ask your system administrator for help, for you cannot run any Java programs before the problem is solved.

Study methods

It can be useful to remember the Confucian saying about learning as you start your studies:

Tell me and I forget;
Show me and I remember;
Let me do it and I understand.

As experts have predicted that more and more people in future will write programs without being programmers, you are recommended to learn the important principles and apply them in your programming practice as much as possible. The experience could be very useful for your future career whatever you do.

More specifically:

1. For every hour of study on new material in lecture, two hours of lab work and two hours of revision or exercises are highly recommended.
2. Use examples to help understanding of each problem.
3. Always ask the question: 'Is there a better solution for the problem?'
4. Practise as much as you can.

Laboratory exercises

There is a one-hour supervised lab session every week for each student.

Lab exercise sheets are set for students to practise their programming skills using the theoretical knowledge gained from the course and will be available soon after lectures each week.

Test and coursework

One test (3%) and one coursework (7%) will be given to help students' studies and preparation for the examination. Any exemption of the test (21 November 2002) or the coursework (deadline: 12 December 2002) will have to be granted by the Senior Tutor (currently Dr Roger Sugden) of the department.

As we have all agreed with the test date and coursework deadline in the first lecture, no request for any further discussion on these two issues to the lecturer is acceptable.

Examination

The content covered in cis212a (term One 2002-3) will be examined in the first half of a three-hour examination. A revision class at the beginning of term Three may be arranged for the examination.

The students will normally be required to answer two out of three questions (to be confirmed), each includes three to five subquestions in each half of the paper. These subsections may be classified as one of the following three types:

- **bookwork** The answers to these questions can be found in the Notes or required chapters of a recommended textbook.
- **similar** The questions are similar to an example in the lecture notes or required chapters of a recommended textbook.
- **unseen** You may have not seen these types of questions before but you should be able to answer them using the knowledge and experience gained from the course.

Every year we advise the examinees to read the questions on the examination paper **carefully**. You should make sure that you fully understand what is required and what subsections are involved in a examination question. You are encouraged to take notes, if necessary, while attempting the questions. Above all, you should be completely familiar with the course material. To achieve a good grade, you need to have prepared well for the examination and to be able to solve problems by applying the knowledge gained from your studies of the course.

Content

The main topics that we plan to cover in the course unit are as below but we may have to adjust them according to students' level:

¹ *To be reviewed*

1

Week 1

Lecture 1-2

The aim, objectives and plan of the course

Self-test: Knowing how much we know about Java

Lab 0 Viewing some useful Web resources

Lab 1 Design short Java programs on arrays

Week 2

Lecture 3-4

Objects, classes and ADTs (lists)

Problem solving, Algorithms Design and Implementation

Lab 2 Identify a problem and solve it in Java programming

Week 3

Lecture 5-6

Recursions

Recursions (lists, trees)

Lab 3 Implementing a recursive program in Java

(2 or 3 questions)

Week 4

Lecture 7-8

Implementation of ADT in Java

Implementation of ADT in Java

Lab 4

Week 5

Lecture 9-10

Java Graphics

Composition, Inheritance, exception, multi-threads

Lab 5

study week

no lectures

Lab 6 at home

Week 7

Lecture 11-12

Design Patterns

Application frameworks 1

Lab 7

Week 8

Lecture 13

Searching

Lecture 14

Test

Lab 8

Week 9

Lecture 15-16

Sorting

Sorting

Lab9

Week 10

Lecture 17-18

Application frameworks 2

Lab 10

Week 11

Lecture 19-20

Revision

Revision

Lab 11

Laboratory Exercise 0

²*Please do this at home
instead of in the Lab.*

VISITING JAVA WEB SITES²

1. Java Books

(a) Thinking in Java

<http://www.brueckeckle.com>

(b) Java Gently

<http://www.cs.up.ac.za/javagently/>

2. What the other first year Java course doing?

(a) David J.Eck's Java course:

<http://math.hws.edu/eck/cs124/>

<http://math.hws.edu/javanotes/>

(b) Java Tutorial

<http://java.sun.com/docs/books/tutorial/>

3. Installing Java system

<http://burks.bton.ac.uk>

<http://java.sun.com/>

<http://textpad.com>

Laboratory Exercise 1

ARRAYS

³*This is an exercise about arrays which you studied in cis109. We need to use arrays for cis212a this term.*

⁴*Q2(c) in the self-test.*

⁵*Q2(d) in the self-test.*

⁶*Q1(e) in the self-test.*

1. ³ ⁴ Define an array to store student marks. Suppose each student has only one mark and there are at most one thousand students in total.
2. ⁵ Following the above, write a method that displays the marks of the students.
3. ⁶ Write a boolean type method which takes (1) an array of integers and (2) the size of the array as parameters and determines if all the integers in the array are between 10 and 50 inclusive.