

A Short Course in Computational Science and Engineering

C++, Java and Octave Numerical Programming with Free Software Tools

Building on his highly successful textbook on C++, David Yevick provides a concise yet comprehensive one-stop course in three key programming languages, C++, Java and Octave (a freeware alternative to MATLAB®).

Employing only public-domain software to ensure straightforward implementation for all readers, this book presents a unique overview of numerical and programming techniques relevant to scientific programming, including object-oriented programming, elementary and advanced topics in numerical analysis, physical system modeling, scientific graphics, software engineering and performance issues. Relevant features of each programming language are illustrated with short, incisive examples, and the installation and application of the software are described in detail. Compact, transparent code in all three programming languages is applied to the fundamental equations of quantum mechanics, electromagnetics, mechanics and statistical mechanics. Uncommented versions of the code that can be immediately modified and adapted are provided online for the more involved programs.

This compact, practical text is an invaluable introduction for students in all undergraduate- and graduate-level courses in the physical sciences or engineering that require numerical modeling, and is also a key reference for instructors and scientific programmers.

DAVID YEVICK is a Professor of Physics at the University of Waterloo. He has been engaged for 30 years in scientific programming in various fields of optical communications and solid state physics at numerous university and industrial establishments, where he performed pioneering work on the numerical modeling of optical communication devices and systems. Professor Yevick is currently a Fellow of the American Physical Society, the Institute of Electrical and Electronics Engineers and the Optical Society of America as well as a registered Professional Engineer (Ontario). He has taught scientific and engineering programming for over 20 years and has authored or co-authored over 170 refereed journal articles.

A Short Course in Computational Science and Engineering

C++, Java and Octave numerical
programming with free software tools

David Yevick



CAMBRIDGE
UNIVERSITY PRESS



CAMBRIDGE
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment,
a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of
education, learning and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9780521116817

© David Yevick 2012

This publication is in copyright. Subject to statutory exception and to the provisions
of relevant collective licensing agreements, no reproduction of any part may take
place without the written permission of Cambridge University Press & Assessment.

First published 2012

A catalogue record for this publication is available from the British Library

Library of Congress Cataloging-in-Publication data

Yevick, David.

A short course in computational science and engineering : C++, Java, and
Octave numerical programming with free software tools / David Yevick.

p. cm.

Includes index.

ISBN 978-0-521-11681-7 (hardback)

1. Computer programming – Textbooks. 2. Computer science – Textbooks. I. Title.

QA76.6.Y48 2012

005.1 – dc23 2011044368

ISBN 978-0-521-11681-7 Hardback

Additional resources for this publication at www.cambridge.org/yevick

Cambridge University Press & Assessment has no responsibility for the persistence
or accuracy of URLs for external or third-party internet websites referred to in this
publication and does not guarantee that any content on such websites is, or will
remain, accurate or appropriate.

אמר רבי יוסי בן קסמא, פעם אחת הייתי מהלך בדרך ופגע בי אדם אחד, ונתן לי שלום, והחזרתי לו שלום, אמר לי, רבי מאיזה מקום אתה, אמרתי לו, מעיר גדולה של חכמים ושל סופרים אני, אמר לי, רבי רצונך שתדור עמנו במקומנו ואני אתן לך אלף אלפים דנרי זהב ואבנים אמרתי לו אם אתה נוהג לי כל כסף וזהב ואבנים, טובות ומרגליות טובות ומרגליות שבעולם, איני דר אלא במקום תורה

Rabbi Yose ben Kisma said: Once I was walking on the road, when a certain man met me. He greeted me and I returned his greeting. He said to me, ‘Rabbi, from what place are you?’ I said to him, ‘I am from a great city of scholars and sages.’ He said to me, ‘Rabbi, would you be willing to live with us in our place? I would give you thousands upon thousands of golden dinars, precious stones and pearls.’ I replied, ‘Even if you were to give me all the silver and gold, precious stones and pearls in the world, I would dwell nowhere but in a place of Torah.’ (Ethics of the Fathers 6:9)

Rabbi Jose, Kisma's son, berättade: En gång gick jag ut och vandrade, då mötte mig en människa, som hälsade mig, och jag besvarade hans hälsning. Han sporde mig: Rabbi, varifrån är du, och jag svarade honom: Från en stor stad, full av visa män och skriftlärda. Då sade han till mig: Rabbi, om du vill bo hos oss i vår stad, vill jag giva dig tusen gånger tusen guldmynt, ädelstenar och pärlor. Jag svarade honom: Om du så gäve mig all världens silver, guld, ädelstenar och pärlor, skulle jag aldrig vilja bo på ett annat ställe än där Torahn har sin hemvist. (Fädernas Tankespråk 6:9)

Contents

1	Introduction	<i>page</i> 1
1.1	Objective	1
1.2	Presentation	1
1.3	Programming languages	2
1.4	Language standards	3
1.5	Chapter summary	4
1.6	How to use this text	4
1.7	Additional and alternative software packages	4
2	Octave programming	5
2.1	Obtaining octave	5
2.2	Command summary	5
2.3	Logistic map	14
3	Installing and running the Dev-C++ programming environment	15
3.1	Compiling and running a first program	15
3.2	The Dev-C++ debugger	17
3.3	Installing DISLIN	19
3.4	A first graphics program	19
3.5	The help system	20
3.6	Example	20
4	Introduction to computer and software architecture	22
4.1	Computational methods	22
4.2	Hardware architecture	23
4.3	Software architecture	24
4.4	The operating system and application software	25
5	Fundamental concepts	26
5.1	Overview of program structure	26
5.2	Tokens, names and keywords	26

5.3	Expressions and statements	27
5.4	Constants, variables and identifiers	27
5.5	Constant and variable types	27
5.6	Block structure	30
5.7	Declarations, definitions and scope	31
5.8	rvalues and lvalues	31
5.9	Operators – precedence and associativity	31
5.10	The const keyword	32
5.11	Formatting conventions	33
5.12	Comments	33
6	Procedural programming basics	35
6.1	Scientific software development	35
6.2	The main() function	36
6.3	Namespaces	37
6.4	Preprocessor directives and #include statements	38
6.5	Arithmetic and logical operators	39
6.6	The bool and enum types	40
6.7	Control flow, if statements and implicit blocks	40
6.8	The for statement	42
6.9	while and do . . . while statements	42
6.10	The break , continue and exit() statements	43
6.11	The typedef keyword	44
6.12	Input and output streams	44
6.13	File streams	45
6.14	Casts	45
6.15	Functions	46
6.16	Principles of function operation	46
6.17	Function declarations and prototypes	48
6.18	Enumerators and functions	48
6.19	Overloading and argument conversion	48
6.20	Built-in functions and header files	49
6.21	The assert statement and try and catch blocks	50
6.22	Multiple return statements	51
6.23	Default parameters	52
6.24	Functions and global variables	52
6.25	Inline functions	53
6.26	Recursive functions	53
6.27	Modular programming	54

6.28	Arrays	54
6.29	Program errors	55
6.30	Numerical errors with floating-point types	56
7	An introduction to object-oriented analysis	58
7.1	Procedural versus object-oriented programming	58
7.2	Problem definition	60
7.3	Requirements specification	61
7.4	UML diagrams	61
7.5	Classes and objects	61
7.6	Object discovery	63
7.7	Inheritance	65
8	C++ object-oriented programming syntax	66
8.1	Class declaration	66
8.2	Class definition and member functions	66
8.3	Object creation and polymorphism	68
8.4	Information hiding	70
8.5	Constructors	71
8.6	Examples	73
8.7	Wrapping legacy code	75
8.8	Inheritance	76
8.9	The “protected” keyword	78
8.10	Multifile programs	78
8.11	const member functions	81
9	Arrays and matrices	83
9.1	Data structures and arrays	83
9.2	Array definition and initialization	83
9.3	Array manipulation and memory access	84
9.4	Arrays as function parameters	86
9.5	Returning arrays as objects and object arrays	87
9.6	const arrays	88
9.7	Multidimensional arrays	89
9.8	Multidimensional array storage and loop order	90
9.9	Multidimensional arrays as function arguments	91
10	Input and output streams	93
10.1	The iostream class and stream manipulators	93
10.2	File streams	95
10.3	The string class and string streams	97
10.4	The toString() class member	98
10.5	The printf function	99

11	References	101
11.1	Basic properties	101
11.2	References as function arguments	102
11.3	Reference member variables	103
11.4	const reference variables	103
11.5	Reference return values	104
12	Pointers and dynamic memory allocation	106
12.1	Introduction to pointers	106
12.2	Initializing pointer variables	107
12.3	The address-of and dereferencing operators	107
12.4	Uninitialized pointer errors	108
12.5	The const keyword and pointers	109
12.6	Pointer arithmetic	110
12.7	Pointers and arrays	110
12.8	Pointer comparisons	111
12.9	Pointers to pointers and matrices	111
12.10	String manipulation	112
12.11	Static and dynamic memory allocation	113
12.12	Memory leaks	115
12.13	Dangling pointers	115
12.14	Pointers in function blocks	117
12.15	Dynamic memory allocation within functions	118
12.16	Dynamically allocated matrices	119
12.17	Dynamically allocated matrices as function arguments	120
12.18	Pointer data structures and linked lists	121
13	Memory management	123
13.1	The this pointer	123
13.2	The friend keyword	124
13.3	Operators	125
13.4	Destructors	127
13.5	Assignment operators	128
13.6	Copy constructors	130
14	The static keyword, multiple and virtual inheritance, templates and the STL	132
14.1	Static variables	132
14.2	Static class members	132
14.3	Virtual functions	134
14.4	Heterogeneous object collections and runtime type identification	135
14.5	Abstract base classes and interfaces	136

14.6	Multiple inheritance	137
14.7	Virtual inheritance	138
14.8	User-defined conversions	138
14.9	Function templates	139
14.10	Templates and classes	140
14.11	The complex class	142
14.12	The standard template library	143
14.13	Structures, unions and nested classes	147
14.14	Bit-fields and operators	148
14.15	Program optimization	149
15	Creating a Java development environment	152
15.1	Basic setup	152
15.2	Command-line operation	153
15.3	A first graphical Java program	155
15.4	DISLIN applet	155
15.5	Graphics applet	156
15.6	Packages	157
15.7	Static (instance) and class members	159
16	Basic Java programming constructs	161
16.1	Comments	161
16.2	Primitive types	161
16.3	Conversions	163
16.4	Operators	164
16.5	Control logic	165
16.6	Enumerations	167
17	Java classes and objects	168
17.1	Class definition	168
17.2	Inheritance	170
17.3	Java references and functions	171
17.4	Exceptions	173
17.5	Basic Java reference types	174
17.6	Input and ouput	177
17.7	File I/O	178
18	Advanced Java features	179
18.1	Dynamic method dispatch	179
18.2	Abstract classes	180
18.3	Interfaces	180
18.4	Java event handling	182
18.5	Multithreading	184

18.6	Serialization	186
18.7	Generic types	186
19	Introductory numerical analysis	188
19.1	The derivative operator	188
19.2	Error dependence	190
19.3	Graphical error analysis	190
19.4	Analytic error analysis – higher-order methods	192
19.5	Extrapolation	193
19.6	The derivative calculator class	193
19.7	Integration	194
19.8	Root-finding procedures	196
19.9	Minimization	198
20	Linear algebra	200
20.1	Matrices	200
20.2	Linear-equation solvers	200
20.3	Errors and condition numbers	204
20.4	Application: least-squares procedure	204
20.5	Eigenvalues and iterative eigenvalue solvers	206
21	Fourier transforms	208
22	Differential equations	212
22.1	Euler’s method	212
22.2	Error analysis	214
22.3	The Runge–Kutta procedure	216
23	Monte Carlo methods	218
23.1	Monte Carlo integration	218
23.2	Monte Carlo evaluation of distribution functions	219
23.3	Importance sampling	220
23.4	The Metropolis algorithm	221
23.5	Multicanonical methods	224
23.6	Particle simulations	225
23.7	The Ising model	227
24	Partial differential equations	230
24.1	Scientific applications	230
24.2	Direct solution methods	234
24.3	Hyperbolic differential equations and electromagnetics	235
24.4	Elliptic equations	240
24.5	Split-operator methods for parabolic differential equations	242

	Contents	xiii
24.6 Symplectic evolution operators in classical mechanics	245	
24.7 Fast Fourier transform methods in optics	246	
24.8 The Crank–Nicholson method in quantum mechanics	252	
24.9 Finite-difference and finite-element procedures	254	
<i>Index</i>	259	