

《计算物理和C++面向对象的程序设计教程》

图书基本信息

书名：《计算物理和C++面向对象的程序设计教程》

13位ISBN编号：9787510087732

出版时间：2015-3-1

作者：David Yevick

页数：402

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

《计算物理和C++面向对象的程序设计教程》

内容概要

《计算物理和C++面向对象的程序设计教程》是C++语言的初学者或者中等水平的编程人员的最佳选择，内容从C++应用到技巧问题，运用现代面向对象软件工程工具以简化表示和现代C++编程有关的科学编程的众多方面。由于这本教程的独特性，对于计算科学的学生和科学编程人员，其作用不可估量。尤其值得关注大篇幅的附录，这些附录都具有极强的应用性，不可忽视。

《计算物理和C++面向对象的程序设计教程》

作者简介

David Yevick，是国际知名学者，在数学和物理学界享有盛誉。本书凝聚了作者多年科研和教学成果，适用于科研工作者、高校教师和研究生。

书籍目录

Part C++programmingbasics

1 Introduction

1.1 Objective

1.2 Presentation

1.3 WhyC++

1.4 C++standards

1.5 Summary

1.6 How to use this text

1.7 Additional study aids

1.8 Additionaland alternative software packages

2 Installing and running the Dev—C++ programming environment

2.1 Compiling and running a first program

2.2 Using the Dev—C++debugger

2.3 Installing DILIN and gsl

2.4 A first graphics program

2.5 The help system

2.6 Linuxalternatives

2.7 Assignment

3 Introduction to computer and software architecture

3.1 Computationalmethods

3.2 Hardware architecture

3.3 Softwarearchitecture

3.4 The operating system and application software

3.5 Assignments

4 Fundamentalconcepts

4.1 Overview ofprogram structure

4.2 Tokens , names , and keywords

4.3 Expressions and statements

4.4 Constants , variables , and identifiers

4.5 Declarations , definitions , and scope

4.6 rvalues and lvalues

4.7 Block structure

4.8 The const keyword

4.9 Operators—precedence and associativity

4.10 Formatting conventions

4.11 Comments

4.12 Assignments

5 Writing a first program

5.1 The main () function

5.2 Namespaces

5.3 #include Statements

5.4 Input and output streams

5.5 File streams

5.6 Constant and variable types

5.7 Casts

5.8 Operators

5.9 Controlflow

- 5.10 Functions
- 5.11 Arrays and typedefs
- 5.12 A first look at scientific software development
- 5.13 Program errors
- 5.14 Numerical errors with floating—point types
- 5.15 Assignments
- 6 An introduction to object—oriented analysis
 - 6.1 Procedural versus object—oriented programming
 - 6.2 Problem definition
 - 6.3 Requirements specification
 - 6.4 UML diagrams
 - 6.5 Use case diagram
 - 6.6 Classes and objects
 - 6.7 Object discovery
 - 6.8 Sequence and collaboration diagrams
 - 6.9 Aggregation and association
 - 6.10 Inheritance
 - 6.11 Object—oriented programming approaches
 - 6.12 Assignments
- 7 C++ object—oriented programming syntax
 - 7.1 Class declaration
 - 7.2 Class definition and member functions
 - 7.3 Object creation and polymorphism
 - 7.4 Information hiding
 - 7.5 Constructors
 - 7.6 Wrapping legacy code
 - 7.7 Inheritance
 - 7.8 The 'protected' keyword
 - 7.9 Assignments
- 8 Control logic and iteration
 - 8.1 The bool and enum types
 - 8.2 Logical operators
 - 8.3 if statements and implicit blocks
 - 8.4 else , else if , conditional and switch statements
 - 8.5 The exit () function
 - 8.6 Conditional compilation
 - 8.7 The for statement
 - 8.8 while and do...while statements
 - 8.9 The break and continue statements
 - 8.10 Assignments
- 9 Basic function properties
 - 9.1 Principles of function operation
 - 9.2 Function declarations and prototypes
 - 9.3 Overloading and argument conversion
 - 9.4 Built—in functions and header files
 - 9.5 Program libraries
 - 9.6 Function preconditions and postconditions—the assert statement
 - 9.7 Multiple return statements
 - 9.8 Functions and global variables

- 9.9 Use of const in functions
- 9.10 Default parameters
- 9.11 Inline functions
- 9.12 Modular programming
- 9.13 Recursive functions
- 9.14 Assignments
- 10 Arrays and matrices
- 10.1 Data structures and arrays
- 10.2 Array definition and initialization
- 10.3 Array manipulation and memory access
- 10.4 Arrays as function parameters
- 10.5 Returning arrays and object arrays
- 10.6 const arrays
- 10.7 Matrices
- 10.8 Matrix storage and loop order
- 10.9 Matrices as function arguments
- 10.10 Assignments
- 11 Input and output streams
- 11.1 The iostream class and stream manipulators
- 11.2 File streams
- 11.3 The string class and string streams
- 11.4 The toString () class member
- 11.5 The printf function
- 11.6 Assignments
- Part Numerical analysis
- 12 Numerical error analysis—derivatives
- 12.1 The derivative operator
- 12.2 Error dependence
- 12.3 Graphical error analysis
- 12.4 Analytic error analysis—higher—order methods
- 12.5 Extrapolation
- 12.6 The derivative calculator class
- 12.7 Assignments
- 13 Integration
- 13.1 Discretization procedures
- 13.2 Implementation
- 13.3 Discretization error
- 13.4 Assignments
- 14 Root—finding procedures
- 14.1 Bisection method
- 14.2 Newton's method
- 14.3 Assignments
- 15 Differential equations
- 15.1 Euler's method
- 15.2 Error analysis
- 15.3 The spring class
- 15.4 Assignments
- 16 Linear algebra
- 16.1 Linear equation solvers

16.2 Errors and condition numbers

16.3 Eigenvalues and iterative eigenvalue solvers

16.4 Assignments

.....

Part III Advanced object-oriented programming

17 References

18 Pointers and dynamic memory allocation

19 Advanced memory management

20 The static keyword; multiple and virtual inheritance, templates, and the STL library

21 Program optimization in C++

Part IV Scientific programming examples

22 Monte Carlo methods

Appendix A Overview of MArLAB

Appendix B The Borland C++ Compiler

Appendix C The Linux / Windows Command-Line C++ Compiler and Profiler

Appendix D Calling FORTRAN programs from C++

Appendix E C++ coding standard

References and further reading

Index

《计算物理和C++面向对象的程序设计教程》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com