

C++ Programming (Li Zheng et al. 2019) - Study Notes

Isaac Riley

Last edited: January 30, 2023

Contents

1	Introduction	3
1.1	The Development of Computer Programming Languages	3
1.2	Object-Oriented Method	3
1.3	Object-Oriented Software Development	3
1.4	Representation and Storage of Information	3
1.5	The Development and Process of Programs	3
1.6	Summary	3
2	Elementary C++ Programming	4
2.1	An Overview of C++ Language	4
2.2	Basic Data Types and Expressions	4
2.3	Data Input and Output	4
2.4	The Fundamental Control Structures of Algorithms	4
2.5	User-Defined Data Type	4
2.6	Summary	4
3	Functions	5
3.1	Definition and Use of Function	5
3.2	Inline Functions	5
3.3	Default Formal Parameters in Functions	5
3.4	Function Overloading	5
3.5	Using C++ System Functions	5
3.6	Summary	5
4	Class and Object	6
4.1	Basic Features of Object-Oriented Design	6
4.2	Class and Object	6
4.3	Constructor and Destructor	6
4.4	Combination of Classes	6
4.5	UML	6
4.6	Program Instance – Personnel Information Management Program	6
4.7	Summary	6
5	Data Sharing and Protecting	7
5.1	Scope and Visibility of Identifiers	7
5.2	Lifetime of Object	7
5.3	Static Members of Class	7
5.4	Friend of Class	7
5.5	Protection of Shared Data	7
5.6	Multifile Structure and Compilation Preprocessing Directives	7
5.7	Example – Personnel Information Management Program	7
5.8	Summary	7

6	Arrays, Pointers, and Strings	8
6.1	Arrays	8
6.2	Pointers	8
6.3	Dynamic Memory Allocation	8
6.4	Deep Copy and Shallow Copy	8
6.5	Strings	8
6.6	Program Example – Personnel Information Management Program	8
6.7	Summary	8
7	Inheritance and Derivation	9
7.1	Inheritance and Derivation of Class	9
7.2	Access Control	9
7.3	Type Compatible Rule	9
7.4	Constructor and Destructor of Derived Class	9
7.5	Identification and Access of Derived-Class Member	9
7.6	Program Example: Solving Linear Equations using Gaussian Elimination Method	9
7.7	Program Example: Personnel Information Management Program	9
7.8	Summary	9
8	Polymorphism	10
8.1	An Overview of Polymorphism	10
8.2	Operator Overload	10
8.3	Virtual Function	10
8.4	Abstract Classes	10
8.5	Program Instance: Variable Stepwise Trapezoid Method to Calculate Functional Definite Integral	10
8.6	Program Instance: Improvement on Staff Information Management System for a Small Corporation	10
8.7	Summary	10
9	Collections and Their Organization	11
9.1	Function Templates and Class Templates	11
9.2	Linear Collection	11
9.3	Organizing Data in Linear Collections	11
9.4	Application – Improving the HR Management Program of a Small Company	11
9.5	Summary	11
10	Generic Programming and the Standard Template Library	12
10.1	Generic Programming	12
10.2	Containers in STL	12
10.3	Iterators	12
10.4	Algorithms in STL	12
10.5	Function Objects	12
10.6	Application – Improving the HR Management Program of a Small Company	12
10.7	Summary	12
11	The I/O Stream Library and Input/Output	13
11.1	I/O Stream’s Concept and the Structure of a Stream Library	13
11.2	Output Stream	13
11.3	Input Stream	13
11.4	Input/output Stream	13
11.5	Example-improve Employee Information Management System	13
11.6	Summary	13
12	Exception Handling	14
12.1	Basic Concepts of Exception Handling	14
12.2	The Implementation of Exception Handling in C++	14
12.3	Destruction and Construction in Exception Handling	14
12.4	Exception Handling of Standard Library	14
12.5	Program Example Improvement to Personal Information Administration Program in a Small Company	14
12.6	Summary	14

1 Introduction

1.1 The Development of Computer Programming Languages

- **Interesting phrasing:** assembly language and high-level languages as “closing the gap” between computers and machines
- Dichotomy: static vs dynamic features

1.2 Object-Oriented Method

Structured programming:

- refinement of procedural programming
- two main ideas:
 1. top-down structure
 2. stepwise refinement
- 3 basic structures:
 1. sequential structure
 2. branch structure
 3. loop structure
- **object: data and its operations as an interdependent and inseparable whole**
- **class: abstraction of object; class : object \sim mold : cast**
- **encapsulation: hide certain aspects; “show” only what is necessary; provide clean interface for interacting with the object**
- **inheritance: make it possible to create a new class from an old class, which will inherit the characteristics of the old (base) class; facilitates reusability**
- **polymorphism: allowing for variability in object characteristics across inheritance**

1.3 Object-Oriented Software Development

abbreviations:

- OOA: object-oriented analysis
- OOD:
- OOP:
- OOT:
- OOSM:
-

1.4 Representation and Storage of Information

1.5 The Development and Process of Programs

1.6 Summary

2 Elementary C++ Programming

2.1 An Overview of C++ Language

2.2 Basic Data Types and Expressions

2.3 Data Input and Output

2.4 The Fundamental Control Structures of Algorithms

2.5 User-Defined Data Type

2.6 Summary

3 Functions

3.1 Definition and Use of Function

3.2 Inline Functions

3.3 Default Formal Parameters in Functions

3.4 Function Overloading

3.5 Using C++ System Functions

3.6 Summary

4 Class and Object

4.1 Basic Features of Object-Oriented Design

4.2 Class and Object

4.3 Constructor and Destructor

4.4 Combination of Classes

4.5 UML

4.6 Program Instance – Personnel Information Management Program

4.7 Summary

5 Data Sharing and Protecting

5.1 Scope and Visibility of Identifiers

5.2 Lifetime of Object

5.3 Static Members of Class

5.4 Friend of Class

5.5 Protection of Shared Data

5.6 Multifile Structure and Compilation Preprocessing Directives

5.7 Example – Personnel Information Management Program

5.8 Summary

6 Arrays, Pointers, and Strings

6.1 Arrays

6.2 Pointers

6.3 Dynamic Memory Allocation

6.4 Deep Copy and Shallow Copy

6.5 Strings

6.6 Program Example – Personnel Information Management Program

6.7 Summary

7 Inheritance and Derivation

7.1 Inheritance and Derivation of Class

7.2 Access Control

7.3 Type Compatible Rule

7.4 Constructor and Destructor of Derived Class

7.5 Identification and Access of Derived-Class Member

7.6 Program Example: Solving Linear Equations using Gaussian Elimination Method

7.7 Program Example: Personnel Information Management Program

7.8 Summary

8 Polymorphism

8.1 An Overview of Polymorphism

8.2 Operator Overload

8.3 Virtual Function

8.4 Abstract Classes

8.5 Program Instance: Variable Stepwise Trapezoid Method to Calculate Functional Definite Integral

8.6 Program Instance: Improvement on Staff Information Management System for a Small Corporation

8.7 Summary

9 Collections and Their Organization

9.1 Function Templates and Class Templates

9.2 Linear Collection

9.3 Organizing Data in Linear Collections

9.4 Application – Improving the HR Management Program of a Small Company

9.5 Summary

10 Generic Programming and the Standard Template Library

10.1 Generic Programming

10.2 Containers in STL

10.3 Iterators

10.4 Algorithms in STL

10.5 Function Objects

10.6 Application – Improving the HR Management Program of a Small Company

10.7 Summary

11 The I/O Stream Library and Input/Output

11.1 I/O Stream's Concept and the Structure of a Stream Library

11.2 Output Stream

11.3 Input Stream

11.4 Input/output Stream

11.5 Example-improve Employee Information Management System

11.6 Summary

12 Exception Handling

12.1 Basic Concepts of Exception Handling

12.2 The Implementation of Exception Handling in C++

12.3 Destruction and Construction in Exception Handling

12.4 Exception Handling of Standard Library

12.5 Program Example Improvement to Personal Information Administration Program in a Small Company

12.6 Summary