

Software Development: Structured Programming: An Introduction

HIGHER NATIONAL DIPLOMA

软件开发：结构化编程（初级）


【英】苏格兰学历管理委员会（SQA）


Unit Student Guide

COMPUTING: Software Development

DH3E 35



 **中国时代经济出版社**


SCOTTISH
QUALIFICATIONS
AUTHORITY

Software Development: Structured Programming: An Introduction

HIGHER NATIONAL DIPLOMA

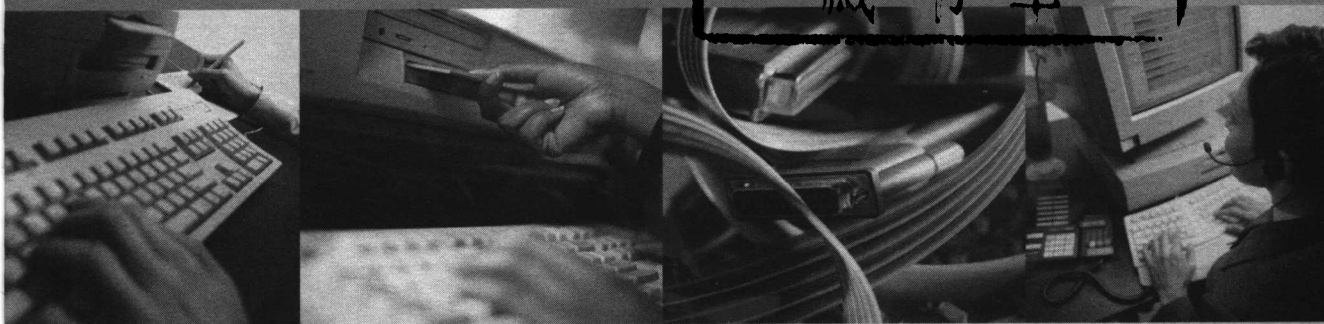
软件开发：结构化编程（初级）

【英】苏格兰学历管理委员会（SQA）

Unit 1 Student Guide

COMPUTING: Software Development

藏书章 35



中国时代经济出版社

SCOTTISH
QUALIFICATIONS
AUTHORITY



著作权合同登记 图字：01-2005-4091号

图书在版编目 (CIP) 数据

软件开发：结构化编程. 初级/苏格兰学历管理委员会著. -北京：中国时代经济出版社，2005.8

ISBN 7-80169-917-3

I. 软… II. 苏… III. 软件开发-教材-英文 IV. TP311.52

中国版本图书馆CIP数据核字 (2005) 第048209号

“First published by CMEPH”

“All Rights Reserved”

“Authorized Apograph/ Translation/Adaptation of the editions by the Scottish Qualifications Authority. All Intellectual Property Rights vest in the Scottish Qualifications Authority and no part of these “Works” may be reproduced in any form without the express written permission of Scottish Qualifications Authority”

Software Development: Structured Programming: An Introduction

软件开发：结构化编程（初级）

苏格兰学历管理委员会著

| | |
|------|---|
| 出版者 | 中国时代经济出版社 |
| 地址 | 北京市东城区东四十条24号 青蓝大厦东办公区11层 |
| 邮政编码 | 100007 |
| 电话 | (010) 68320825 (发行部) (010) 88361317 (邮购) |
| 传真 | (010) 68320634 |
| 发行 | 各地新华书店 |
| 印刷 | 北京鑫海达印刷有限公司 |
| 开本 | 787×1092 1/16 |
| 版次 | 2005年8月第1版 |
| 印次 | 2005年8月第1次印刷 |
| 印张 | 15 |
| 定价 | 38.00元 |
| 书号 | ISBN 7-80169-917-3/G·256 |

版权所有 侵权必究

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction to the Scottish Qualifications Authority | 1 |
| 2 | Introduction to the Unit | 3 |
| 2.1 | What is the Purpose of this Unit? | 3 |
| 2.2 | What are the Outcomes of this Unit? | 3 |
| 2.3 | What do I Need to be Able to do in Order to Achieve this Unit? | 4 |
| 2.4 | Approximate Study Time for This Unit | 4 |
| 2.5 | Equipment/Material Required for this Unit | 4 |
| 2.6 | Symbols Used in this Unit | 5 |
| 3 | Assessment Information for this Unit | 9 |
| 3.1 | What Do I Have to Do to Achieve This Unit? | 9 |
| 4 | Suggested Lesson Plan | 11 |
| 5 | Learning Material | 13 |
| 5.1 | Creating and Running C++ Programs | 13 |
| 5.2 | Simple Output Statements | 20 |
| 5.3 | Style | 26 |

| | | |
|-------------------|--|------------|
| 5.4 | Comments | 27 |
| 5.5 | Variables | 31 |
| 5.6 | Doing Some Simple Arithmetic | 44 |
| 5.7 | Using Floating Point Variables | 50 |
| 5.8 | Input Statements | 54 |
| 5.9 | Repetition — Using FOR Loops | 61 |
| 5.10 | Simple Selection | 77 |
| 5.11 | Named Constants | 101 |
| 5.12 | Typecasts | 109 |
| 5.13 | DO Loops | 112 |
| 5.14 | Complex Selection | 130 |
| 5.15 | Input Validation | 150 |
| 5.16 | WHILE Loops | 155 |
| | | |
| 6 | Additional Reading Material | 167 |
| | | |
| 7 | Solutions to Self-Assessed Questions and Activities | 169 |
| | | |
| 8 | Copyright References | 209 |
| | | |
| 9 | Acknowledgements | 211 |
| | | |
| Appendix 1 | — Unit Specification | 213 |

1

Introduction to the Scottish Qualifications Authority

This Unit **DH3E 35 — Software Development: Structured Programming** has been devised and developed by the Scottish Qualifications Authority (SQA). Here is an explanation of the SQA and its work:

The SQA is the national body in Scotland responsible for the development, accreditation, assessment and certification of qualifications other than degrees.

Its website can be viewed on: www.sqa.org.uk.

SQA's functions are to:

- devise, develop and validate qualifications, and keep them under review
- accredit qualifications
- approve education and training establishments as being suitable for entering people for these qualifications
- arrange for, assist in, and carry out, the assessment

of people taking SQA qualifications

- quality assure education and training establishments which offer SQA qualifications
- issue certificates to candidates.

In order to pass SQA Units, students must complete prescribed assessments. These assessments must meet certain standards.

The Unit Specification outlines the **four** Outcomes that students must complete in order to achieve this Unit. The Specification also details the knowledge and/or skills required to achieve the outcome or outcomes. The Evidence Requirements prescribe the type, standard and amount of evidence required for each outcome or outcomes.

2

Introduction to the Unit

2.1

What is the Purpose of this Unit?

This Unit is designed to help you develop a broad knowledge of the concepts, principles, boundaries and scope of software development using a structured programming language. This will be reinforced by helping you to develop the practical skills required in using the structures and features of a procedural programming language in the creation of software solutions to problems. It forms part of an HN Computing Group Award programme, although it can also be used as a stand-alone Unit if you wish to acquire and develop programming skills using a procedural language.

2.2

What are the Outcomes of this Unit?

There are 4 Outcomes:

1. Use programming techniques to develop program modules
2. Implement a solution from a design
3. Test the completed product
4. Create technical and user documentation

Further details can be found in Appendix 1 — Unit Specifications.

2.3

What do I
Need to be
Able to do in
Order to
Achieve this
Unit?

This Unit is in two parts. You should work through the materials in this Student Guide and the second Student Guide. You should tackle the Activities and Self-Assessed Questions seriously, trying to complete them before referring to the answers in Section 7.

You should also consult your tutor when the need arises, using his/her knowledge and expertise to help you when you find a topic difficult.

2.4

Approximate
Study Time
for This Unit

Completion of this Unit is intended to be flexible.

The notional study time for this Unit is 80 hours but actual time allocated is at the discretion of the centre.

2.5

Equipment/
Material
Required for
this Unit

You will need access to a reasonably current PC with at least a C++ compiler and linker. Most setups these days consist of an Integrated Development Environment (Editor, Compiler and Linker) which makes the process of creating and developing programs much easier and more user-friendly.

If possible, try to ensure that your compiler conforms to ISO/ANSI standards for C++. Borland C++ and Microsoft C++ are two possible contenders, but it is

also possible to find free C ++ compilers etc on the Internet.

A word processor such as Microsoft Word would be useful for producing the evidence for some of the outcomes.

You will need to complete the following Unit Student Guides to meet all the requirements of this unit:

- Unit Student Guide — Software Development: Structured Programming: An Introduction
- Unit Student Guide — Software Development: Structured Programming: Advanced.

2.6 Symbols Used in this Unit

The various Learning Materials sections are designed so that you can work at your own pace, with tutor support. As you work through the Learning Materials (see Section 5), you will encounter symbols. These symbols indicate that you are expected to do a task. **These tasks are not Outcome Assessments.** They are exercises designed to consolidate learning or encourage thought, in preparation for the Outcome Assessment (see Section 3 — Assessment Information for this Unit).

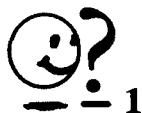
Activity



This symbol indicates an Activity. Usually, Activities are used to improve or consolidate your understanding of the subject in general or a particular feature of it.

In this Unit, you are asked to undertake a number of programming Activities — creating programs, suggesting changes etc.

Self-Assessed Question



This symbol indicates a Self-Assessed Question. Using a Self-Assessed Question helps you check your understanding of the content that you have already covered.

Everything is provided for you to check your own responses. Answers to the Self-Assessed Questions and Activities are to be found at the back of the Unit Student Guide. Where suggested responses to Self-Assessed Question and Activities are provided in the Unit Student Guide, **students are strongly discouraged from looking at these responses before they attempt the activity or question.** The

Self-Assessed Questions and Activities throughout the Unit Student Guide will help you to prepare yourself for the formal assessments, and to identify topic areas in which you will require clarification and additional tutor support. The Self-Assessed Questions and Activities will not serve this purpose if you look at the answers before trying them!

Self-Assessed Questions and Activities are designed to be checked by you. No tutor input is necessary at this stage unless special help is requested, although from time to time your tutor may wish to view your responses to Self-Assessed Questions to see how you are progressing.

3

Assessment Information for this Unit

3.1

What Do I
Have to Do
to Achieve
This Unit?

There are two assessments for this Unit:

1. The main assessment is a project that covers all four outcomes. You will have to write a substantial piece of software from a given design, test the completed software and create technical and user documentation for the software.
2. This assessment is a closed-book theory test. It consists of 20 questions covering programming techniques and concepts. You must answer at least 60% of the questions correctly.

It is possible that your Centre may integrate the assessment of this Unit with that of Unit Software Development: Planning.

4

Suggested Lesson Plan

The Learning Materials (see Section 5) are designed to lead you through a series of Activities that will allow you to consolidate your learning and check on your own progress. Please follow the order of the exercises as they are laid out in Section 5.

Activities and Self-Assessed Questions are placed throughout the Learning Materials at relevant points. Suggested solutions to both are given in Section 7, although **you are strongly advised not to look at these until you have attempted to answer each question or tried out each practical exercise first.**

Some of the Activities are headed 'If you have time'. These activities are optional and you can tackle them if you feel you are ahead and have time to spare. It may be prudent to ask your tutor whether you should tackle them.

The notional length of the course is 80 hours and the two Student Guides are intended to provide learning material to cover this timescale. Although it is difficult to be precise, the split in terms of time between the two Guides is likely to be in the region of:

- Unit Student Guide — Software Development: Structured Programming: An Introduction — 35 hours
- Unit Student Guide — Software Development: Structured Programming: Advanced — 45 hours.

The exact amount of time to be spent on each topic cannot be specified as it will depend on what extra resources your tutor brings to each topic.