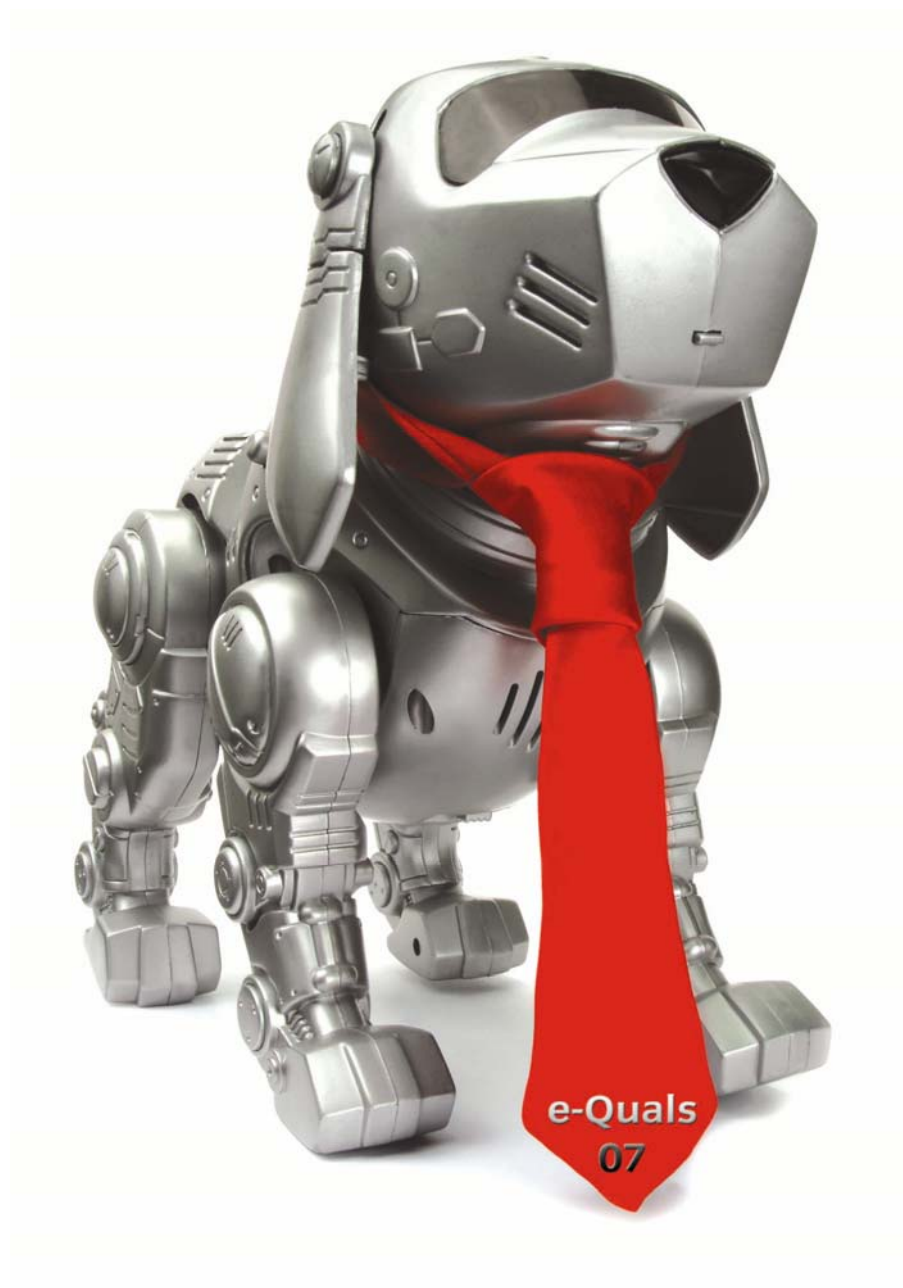


e-Quals Unit Syllabus

Level 2 Create software components using C++
7266/7267 – 202



About City & Guilds

City & Guilds is the UK's leading provider of vocational qualifications, offering over 500 awards across a wide range of industries, and progressing from entry level to the highest levels of professional achievement. With over 8500 centres in 100 countries, City & Guilds is recognised by employers worldwide for providing qualifications that offer proof of the skills they need to get the job done.

City & Guilds Group

The City & Guilds Group includes City & Guilds, ILM (the Institute of Leadership & Management) which provides management qualifications, learning materials and membership services, NPTC which offers land-based qualifications and membership services, and HAB (the Hospitality Awarding Body). City & Guilds also manages the Engineering Council Examinations on behalf of the Engineering Council.

Equal opportunities

City & Guilds fully supports the principle of equal opportunities and we are committed to satisfying this principle in all our activities and published material. A copy of our equal opportunities policy statement *Access to assessment and qualifications* is available on the City & Guilds website.

Copyright

The content of this document is, unless otherwise indicated, © The City and Guilds of London Institute 2007 and may not be copied, reproduced or distributed without prior written consent.

However, approved City & Guilds centres and learners studying for City & Guilds qualifications may photocopy this document free of charge and/or include a locked PDF version of it on centre intranets on the following conditions:

- centre staff may copy the material only for the purpose of teaching learners working towards a City & Guilds qualification, or for internal administration purposes
- learners may copy the material only for their own use when working towards a City & Guilds qualification
- the *Standard Copying Conditions* on the City & Guilds website.

Please note: National Occupational Standards are not © The City and Guilds of London Institute. Please check the conditions upon which they may be copied with the relevant Sector Skills Council.

Publications

City & Guilds publications are available on the City & Guilds website or from our Publications Sales department at the address below or by telephoning +44 (0)20 7294 2850 or faxing +44 (0)20 7294 3387.

Every effort has been made to ensure that the information contained in this publication is true and correct at the time of going to press. However, City & Guilds' products and services are subject to continuous development and improvement and the right is reserved to change products and services from time to time. City & Guilds cannot accept liability for loss or damage arising from the use of information in this publication.

City & Guilds

1 Giltspur Street

London EC1A 9DD

T +44 (0)20 7294 2787

F +44 (0)20 7294 2413

www.cityandguilds.com

centresupport@cityandguilds.com

Contents

Unit 202	Create software components using C++	
Outcome 1	Specify the basic tools required to create, compile and execute a program	4
Outcome 2	Construct and execute a program that uses input and output	5
Outcome 3	Use pre-defined functions	7
Outcome 4	Use operators for arithmetic and logical purposes	8
Outcome 5	Use control structures and functions within a program	10
Outcome 6	Test a software component and produce printed output.	11
Unit record sheet		12

Rationale

The aim of this unit is to enable candidates to understand the principles required to create software using the C++ programming language. Candidates will develop the skills required to create and test software components to solve a given problem.

Learning outcomes

There are **six** outcomes to this unit. The candidate will be able to:

Specify the basic tools required to create, compile and execute a program

Construct and execute a program that uses input and output

Use pre-defined functions

Use operators for arithmetic and logical purposes

Use control structures and functions within a program

Test a software component and produce printed output.

Guided learning hours

It is recommended that 60 hours should be allocated for this unit. This may be on a full time or part time basis.

Connections with other qualifications

This unit contributes towards the knowledge and understanding required for the following qualifications:

Outcome	This award contributes to the knowledge and understanding of the following Areas of Occupational Competence in the City & Guilds NVQ for IT Practitioners (4324)
1,2,3,4,5,6	211 Software Development, component creation 2

Key Skills

This unit contributes towards the Key Skills in the following areas:

Communication	C3.2
Application of Number	N1.1
Information technology	None
Working with others	None
Improving own learning	LP3.1, LP3.2, LP3.3
Problem solving	PS3.1, PS3.2, PS3.3

Assessment and grading

Assessment will be by means of a **set assignment** covering practical activities and a **multiple choice test** covering underpinning knowledge.

Unit 202

Outcome 1

Create software components using C++

Specify the basic tools required to create, compile and execute a program

Practical skills

The candidate will be able to:

- 1 select the tools required to create source code
- 2 create a program using a text editor
- 3 save and retrieve source code to/from disk
- 4 use the standard input/output library
#include <iostream.h>
- 5 compile source code
- 6 interpret and resolve compilation error messages
- 7 edit source code.

Underpinning knowledge

The candidate will be able to:

- 1 identify the need for indentation in code to aid readability
- 2 identify the basic structure of a C++ program
- 3 explain the need for the **#include <iostream.h>** directive
- 4 describe the pre-processor stage
- 5 state the purpose of compiling source code
- 6 describe why object code is linked with library files in producing executable code
- 7 describe the difference between source code, object code and executable (EXE) code
- 8 state which file type can be edited
- 9 state the difference between compilation errors and run-time errors.

Unit 202

Outcome 2

Create software components using C++

Construct and execute a program that uses input and output

Practical skills

The candidate will be able to:

- 1 create a program which defines data types:
 - a **char**
 - b **int**
 - c **float**
 - d **char[]**
- 2 use **cin>>** to read from the keyboard
- 3 use meaningful variable names
- 4 use the **#define** directive to create a symbolic name or constant
- 5 use const to declare constants of type:
 - a **int**
 - b **float**
 - c **char**
- 6 create and use a one-dimensional array of type:
 - a **char**
 - b **int**
 - c **float**
- 7 use the correct syntax to produce screen messages
- 8 use stream manipulators.

Underpinning knowledge

The candidate will be able to:

- 1 identify the correct data types to be used
- 2 state that variable names in C++ are case sensitive
- 3 state the reason for using uppercase for constant names and lowercase for variable and function names
- 4 state the difference between a character variable and a character string when using the symbols " " or ' '
- 5 explain the purpose of the null terminator in relation to a string
- 6 identify the problems in using **cin>>** to input strings (whitespace)
- 7 state the difference between a constant and a variable
- 8 identify the syntax required to declare an array
- 9 state that data types must be compatible with the data being assigned

10 describe the meaning of the following syntax:

- a **#include**
- b **#define**
- c **int main()**
- d **return**
- e **void main()**
- f **cout<<**
- g **cin>>**
- h **;** (semi-colon)
- i **/* */**
- j **//**

11 describe the purpose of the following formatters:

endl \n \r \t \\ \a \" \'

Practical skills

The candidate will be able to:

- 1 use pre-defined functions similar to:
 - a **getch()**
 - b **getche()**
 - c **cin.getline()**
 - d **gets()**
 - e **clrscr()**
 - f **clreol()**
 - g **toupper()**
 - h **tolower()**
- 2 convert a string using:
 - a **atoi()**
 - b **atof()**
- 3 use **strcpy()** to assign a string
- 4 input strings using:
 - a **cin.getline()**
 - b **gets()**

Underpinning knowledge

The candidate will be able to:

- 1 identify the purpose of pre-defined functions
- 2 state the meaning of 'exceeding array bounds' when manipulating strings
- 3 state the purpose of the **strcpy()** function
- 4 state the difference between echoed and non-echoed character input eg **getch()** and **getche()**
- 5 identify the appropriate pre-defined function for converting a string to a numeric value
- 6 identify the data type returned when using **atoi()** and **atof()**.

Unit 202

Outcome 4

Create software components using C++

Use operators for arithmetic and logical purposes

Practical skills

The candidate will be able to:

- 1 use the assignment operator **=** (equals) in a program
- 2 use the **++** and **--** operators in prefix and postfix mode
- 3 use the arithmetic operators:
 - a **ie**
 - b *****
 - c **/**
 - d **-**
 - e **+**
 - f **%**
- 4 use a conditional statement which includes **if** and **else**
- 5 use a **switch** statement
- 6 create simple and compound statements
- 7 use relational operators
- 8 use logical operators:
 - a **ie**
 - b **!** (not)
 - c **&&** (and)
 - d **||** (or)
- 9 use constants to represent **TRUE** and **FALSE**
- 10 use the **ASCII** code as part of validation.

Underpinning knowledge

The candidate will be able to:

- 1 state the order of precedence for arithmetic operators including the use of parenthesis
- 2 state the difference between the **=** and **==** symbols
- 3 identify the role of conditional statements within a program
- 4 state the use of a switch statement
- 5 describe the **++** and **--** operators purpose in prefix and postfix mode
- 6 describe compound statements (eg the **nesting** of **if** statements)
- 7 describe the purpose of the symbols:
 - a **{ }**
 - b **()**

c []

d < >

8 describe the actions of the relational operators:

a <

b <=

c ==

d !=

e >

f >=

Unit 202

Outcome 5

Create software components using C++

Use control structures and functions within a program

Practical skills

The candidate will be able to:

- 1 use control structures for loops:
 - a **while**
 - b **do...while**
 - c **for**
- 2 use control structures as part of a validation process
- 3 create functions with/without parameters
- 4 use a function to return a value
- 5 demonstrate the difference between local and global variables

Underpinning knowledge

The candidate will be able to:

- 1 identify the purpose and format of the control structures:
 - a **while**
 - b **do...while**
 - c **for**
- 2 describe how control structures can assist in the validation of user input
- 3 state that functions create a modular solution to a program and that **main()** is a function
- 4 explain why function prototypes have to be declared in a program
- 5 explain the difference between **global** and **local** variables
- 6 explain the meaning of **scope** in relation to local and global variables
- 7 state the difference between passing parameters to a function:
 - a by **value** (copy)
 - b by **reference**.

Unit 202

Outcome 6

Create software components using C++

Test a software component and produce printed output.

Practical skills

The candidate will be able to:

- 1 write code that is consistently indented
- 2 add comments to a program
- 3 print a program listing
- 4 use test data to determine the expected results from a software component
- 5 compare the expected results to the actual results and correct any errors
- 6 resolve any logical and run-time errors found during testing
- 7 provide evidence that the program complies with the specification.

Underpinning knowledge

The candidate will be able to:

- 1 explain the need to use meaningful comments to a program to aid understanding of a program
- 2 explain the difference between `/* */` and `//`
- 3 state the benefits of producing a hard copy (program listing) of source code
- 4 identify that testing for expected output can assist in determining whether or not the program is working correctly and conforms to the specification.

Unit record sheet

Use this form to track your progress through this unit.

Tick the boxes when you have covered each outcome. When they are all ticked, you are ready to be assessed.

Outcome	✓	Date
1 Specify the basic tools required to create, compile and execute a program	<input type="checkbox"/>	
2 Construct and execute a program that uses input and output	<input type="checkbox"/>	
3 Use pre-defined functions	<input type="checkbox"/>	
4 Use operators for arithmetic and logical purposes	<input type="checkbox"/>	
5 Use control structures and functions within a program	<input type="checkbox"/>	
6 Test a software component and produce printed output.	<input type="checkbox"/>	

Candidate Signature

Date

City & Guilds
Registration Number

Centre Name

Centre Number

Published by City & Guilds

1 Giltspur Street

London

EC1A 9DD

T +44 (0)20 7294 2468

F +44 (0)20 7294 2400

www.cityandguilds.com

www.cityandguilds.com/e-quals07

**City & Guilds is a registered charity
established to promote education and
training**