

BCS Higher Education Qualifications

Level 6 Programming Paradigms Syllabus

Version 3.0

December 2016

This is a United Kingdom government regulated qualification which is administered and approved by one or more of the following: Ofqual, Qualification in Wales or SQA.

Contents

1. Change History	3
2. Rationale	3
3. Aims	3
4. Objectives	3
5. Prior Knowledge Expected	4
6. Format and Duration of the Examination	4
7. Syllabus Detail	5
8. Recommended Reading List	6
9. Contact Points	6

1. Change History

Any changes made to the syllabus shall be clearly documented with a change history log. This shall include the latest version number, date of the amendment and the changes made. The purpose is to identify quickly what changes have been made.

Version Number	Date	Changes Made
Version 1.0	Feb 2014	Previous Release.
Version 2.0	May 2016	Updated to new format. Corrections to some of the ISBN details. Minor changes to join some content together in categories 2 and 5.
Version 3.0	Dec 2016	Regulation statement added.

2. Rationale

Much software has been developed in procedural programming languages which make use of similar control constructs and which differ only in the application area with which the software is associated. The modern programmer, however, may choose from a wide selection of languages, each of which offers a new perspective on the task of software construction. This module aims to provide the candidate with an overview of modern programming languages and the programming paradigms they implement.

3. Aims

- To explore a range of modern programming languages and programming techniques.
- To appreciate the contribution language designers can make to software engineering practice.
- To select appropriate software development tools for given application environments.

4. Objectives

Upon successful completion of this module, candidates will be able to demonstrate their competence in, and their ability to:

- compare and contrast a range of programming paradigms;
- evaluate programming language features critically with respect to the way they support good software engineering practice;
- discuss the appropriateness of the use of a given programming paradigm within a given environment.

5. Prior Knowledge Expected

Certificate in IT

Candidates are expected to be familiar with the material covered in the Certificate syllabuses.

Diploma in IT

Candidates are expected to be familiar with the material covered in the Diploma syllabus for Object Oriented Programming.

Professional Graduate Diploma in IT

The learner must have achieved the Diploma in IT or have an appropriate exemption to be entered for the Professional Graduate Diploma in IT.

Candidates are required to become a member of BCS, The Chartered Institute for IT to sit and be awarded the qualifications. Candidates may apply for a four year student membership that will support them throughout their studies.

6. Format and Duration of the Examination

The examination is a three hour closed book examination (no materials can be taken into the examination room) based on the syllabus in this document.

Examinations are held once a year and are undertaken in normal examination conditions with one or more duly appointed invigilators.

The pass mark is 40%.

7. Syllabus Detail

Category	Ref	Content
1 THE NATURE OF PROGRAMMING LANGUAGES	1.1	Imperative languages and non-imperative languages
	1.2	Scripting languages
	1.3	Data-oriented languages
	1.4	Object-oriented languages
	1.5	Event-driven Programming
	1.6	Language Standardisation
2 PROGRAMMING ENVIRONMENTS	2.1	Compilers and Interpreters
	2.2	Interactive development tools
	2.3	Run-time support environments
	2.4	Debugging Tools
	2.5	Testing Tools
	2.6	Configuration Management
3 OBJECT ORIENTATION	3.1	Basic concepts: objects, classes, methods, overloading methods, messages inheritance: overriding methods, single inheritance, multiple inheritance Interfaces (e.g. in Java), encapsulation, polymorphism.
4 FUNCTIONAL PROGRAMMING	4.1	Definition of a function: domain and range, total and partial functions, strict functions.
	4.2	Recursion
	4.3	Referential transparency
	4.4	Side effects of functions
5 LOGIC PROGRAMMING	5.1	Basic constructs
	5.2	Facts: queries, existential queries, conjunctive queries and rules
	5.3	Definition and semantics of a logic program
	5.4	Recursive programming: Computational model of logic programming
	5.5	Goal reduction
	5.6	Negation in logic programming
6 RELATED ISSUES	6.1	Visual Programming
	6.2	Concurrency
	6.3	Distribution

8. Recommended Reading List

Programming Paradigms	ISBN 10	ISBN 13
Primary Texts Any one of the following four texts will provide adequate coverage of most of the syllabus, although they do not cover concurrency and distribution, and the coverage of scripting languages and event-driven programming is very limited.		
• Clark R. G., Comparative Programming Languages, Addison-Wesley (3rd Ed.), 2000.	0201710129	978-0201710120
• Mitchell, J. C. Concepts in Programming Languages, Cambridge University Press, 2002	0521780985	978-0521780988
• Sebesta, R. W., Concepts of Programming Languages, Addison-Wesley (11 th Ed.) 2015	013394302X	978-0133943023
• Tucker A. and Noonan R, Programming Languages: Principles and Paradigms, McGraw Hill (2 nd Ed.) 2006.	0072866098	978-0072866094

9. Contact Points

Email:

Customer Service team via www.bcs.org/contact

Phone:

UK: 01793 417424 or 0845 300 4417 (lo-call rate)

Overseas: +44 (0)1793 417424

Lines are open Monday to Friday, 08.15 a.m. to 5.45 p.m. UK time.

Website:

www.bcs.org/heq

Post:

BCS, The Chartered Institute for IT
First Floor, Block D, North Star House, North Star Avenue,
Swindon SN2 1FA, United Kingdom