

IB9JH-15 Programming for Quantitative Finance

26/27

Department

Warwick Business School

Level

Taught Postgraduate Level

Module leader

Roman Kozhan

Credit value

15

Module duration

19 weeks

Assessment

100% coursework

Study location

University of Warwick main campus, Coventry

Description

Introductory description

This module is driven by links into, and provides foundation for, the MSMF core structure: the latter determines the set of skills required and provides examples for applications. In turn, this module develops and supports the skills required for practical applications of theoretical concepts developed elsewhere in the core.

[Module web page](#)

Module aims

Objectives:

To develop an understanding of the concepts and “way of thinking” of (object-oriented) programming in general, and practical programming ability in the c++ language.

To provide a framework in which theoretical concepts and methodology developed in the MSMF core can be tested and applied to “real world” problems, thus reinforcing both the theoretical concepts as well as practical programming skills.

To develop a set of transferrable skills required/desired by employers in the Quantitative Finance

industry, enhancing students' competitive "edge" in the job market and widening their choice of attainable career destinations.

Outline syllabus

This is an indicative module outline only to give an indication of the sort of topics that may be covered. Actual sessions held may differ.

At the time of writing, we plan to focus on the C++ programming language. Students will gain in-depth training and practice in R and Python in other core modules of the course.

General Principles

- Principles of programming, algorithms
- Common and distinguishing features of different languages
- Guide to "Good Practice" in programming
- Using an Integrated Development Environment (IDE)
- Different workflow models for application development
- Testing and "debugging"

Fundamentals of Programming

- Data types and data structures
- Constants and variables, namespaces and scope
- Program flow control
- Functions
- Input/output
- Exception handling

Object Oriented Programming

- Principles: classes, objects, instances
- Member functions
- Inheritance and polymorphism
- Templates and generic programming
- Design principles

Applications in Quantitative Finance

Learning outcomes

By the end of the module, students should be able to:

- Demonstrate a comprehensive understanding and explanation of the common and distinct features, including syntax, of a variety of programming languages
- Define and explain, both intuitively and formally, the concepts of "object oriented programming" and application design
- Critically evaluate the design choices and testing for coded programming solutions for a particular problem, both for self-written code and code written by others

Indicative reading list

[Reading lists can be found in Talis](#)

Interdisciplinary

This module is part of a course taught in conjunction with the Maths and Stats departments

Subject specific skills

Design, build, and test applications from scratch to accomplish tasks and/or solve real world problems in Quantitative Finance

Identify, acquire and use public code libraries to incorporate in their own applications to achieve tasks as outlined in preceding item

Conceptualise a computational problem to select and specify appropriate data structures and methods to design, implement, and test/validate solutions

Transferable skills

Written communication

Problem solving

Numeracy

Study

Study time

Type	Required
Lectures	13 sessions of 1 hour (9%)
Project supervision	1 session of 1 hour (1%)
Practical classes	(0%)
Other activity	26 hours (17%)
Private study	44 hours (29%)
Assessment	66 hours (44%)
Total	150 hours

Private study description

preparation for lab workshops; preparation for / revision of lectures; self-study (reading, practice)

Other activity description

13 x 2 hours Practical classes

Costs

No further costs have been identified for this module.

Assessment

You do not need to pass all assessment components to pass the module.

Assessment group A3

	Weighting	Study time	Eligible for self-certification
Assessment component			
Individual Programming Assignment	50%	33 hours	Yes (extension)
Reassessment component is the same			
Assessment component			
Individual Report 1500 word individual report	30%	20 hours	Yes (extension)
Reassessment component is the same			
Assessment component			
Group Programming Report A group programming assignment (no word count as code)	20%	13 hours	No
Reassessment component is the same			

Feedback on assessment

Cohort-level feedback (incl. detailed solution notes) will be given after the individual programming assignment. Formative and summative individual written feedback will be given following the individual group project report.

Availability

Courses

Course availability information is based on the current academic year, so it may change.

This module is Core for:

- Year 1 of TIBS-N3G2 Postgraduate Taught Mathematical Finance