Subject Knowledge

**Aims**  This module teaches the principles of C++ programming and the design of algorithms on modern computer architectures.

**Learning Outcomes**  Students should be able to: understand the components of a C++ program, the structures required to write basic algorithms, algorithm analysis and design, modern computer architectures and memory hierarchies and some algorithms for scientific computing.

**Methods**  Class sessions, recommended textbook and worksheets.

**Assessment**  Marked coursework, written examination.

Skills

**Aims**  To develop design, analysis and problem solving skills.

**Learning Outcomes**  Students will be able to mathematically analyse the computing performance of algorithms. They will be able to design algorithms to solve computing problems.

**Methods**  Class sessions together with worksheets.

**Assessment**  Marked coursework and traditional written examination.

Explanation of Prerequisites  It is assumed that students are already familiar with a programming language such as Fortran, Java or C. Students with little previous programming experience will be required to attend 24 additional hours of programming lectures and laboratories at the start of the course.

**Course Description**  Over the past 15 years C++ has become one of the world’s most popular programming languages, due to its potential for producing efficient and compact code. As such any scientist wishing to develop efficient programs should be familiar with the use of its central features. This module is intended to give the student a basic grasp of its use for scientific computing.

Algorithms are traditionally designed, analysed and compared on a theoretical computer model. However an algorithm that is fast on the theoretical model may be relatively slow on modern computers. This module introduces the memory hierarchies on modern computer architectures and its effects on algorithm performance. The module then teaches how to design fast algorithms on modern computers.

**Detailed Syllabus**

- Introduction to C++.
- Classes, constructors and destructors.
- Pointers, arrays and references.
- Methods and operators.
- Overloading.
- Templates and inheritance.
- Introduction to LAPACK++.
- Algorithm analysis and design.
- Modern computer architectures and memory hierarchies.
- Design of algorithms for matrix operations, polynomials and the fast fourier transform.

Reading List


Resources  Course notes, web page, study guide and worksheets.

Module Evaluation  Course questionnaires, course review.