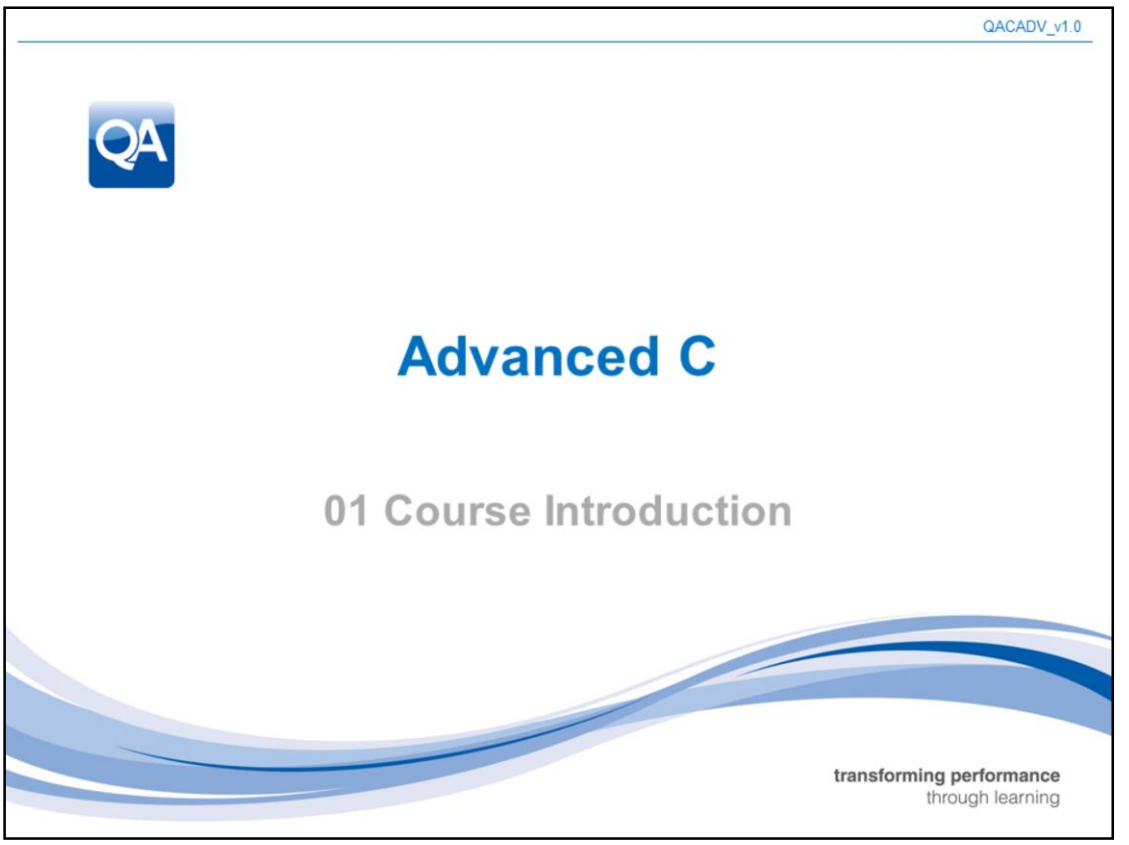
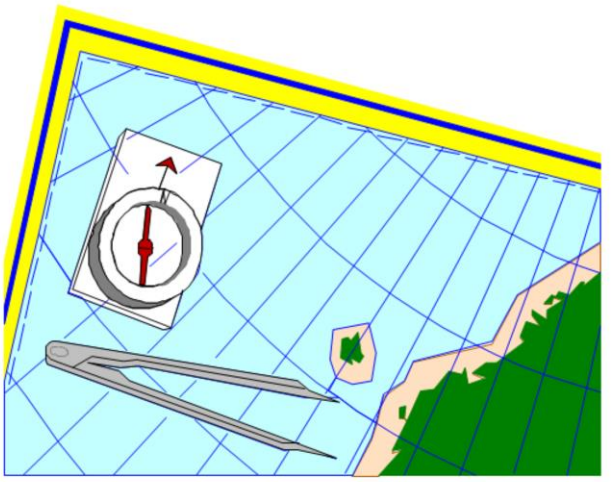

QACADV_v1.0
Advanced C
01 Course Introduction
transforming performance
through learning

QACADV_v1.0

## Course Introduction

- **Objectives**
  - To set the scene and context for the course
- **Contents**
  - Course Prerequisites
  - Course Objectives
  - Course Delivery
  - Course Practicals
  - Course Structure
- **Questions?**



This chapter introduces the Advanced C course and discusses the skills each delegate is expected to have.

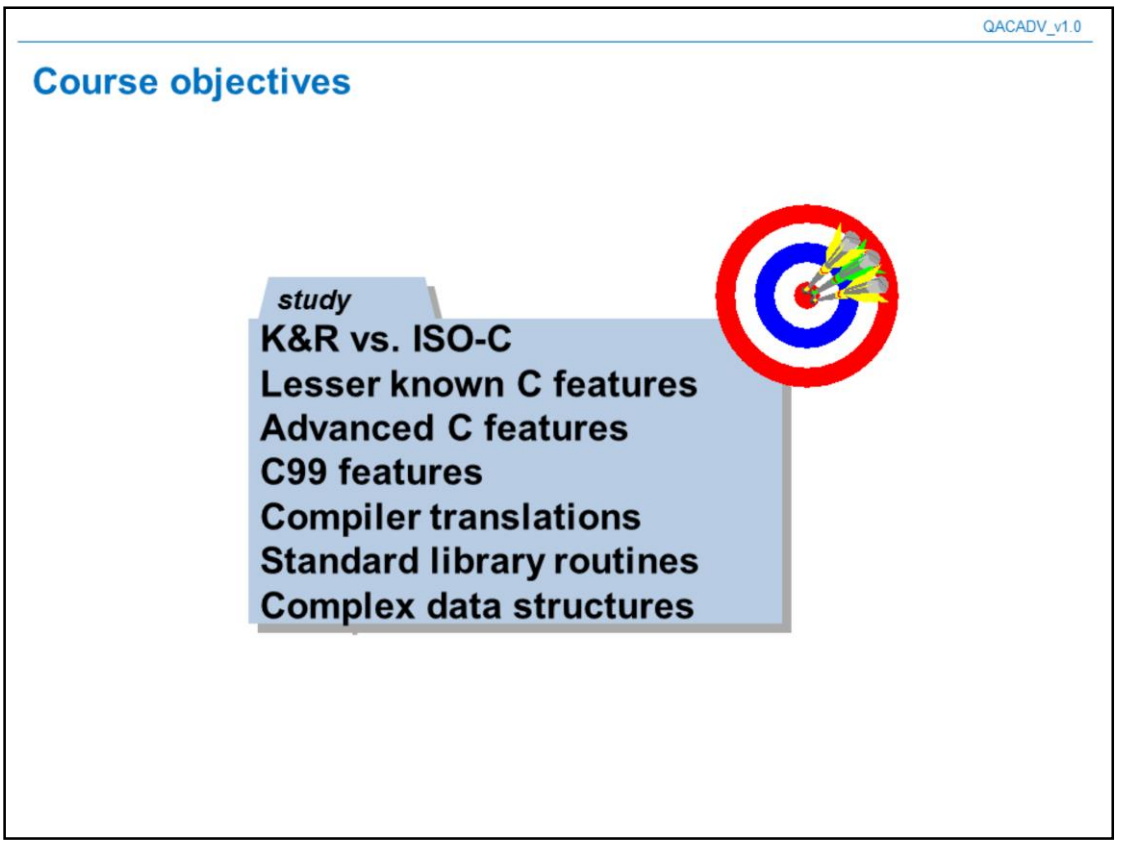
The history of C is discussed, both before and after the American National Standards Institute (ANSI) X3J11 Committee became involved.

QACADV_v1.0

## Course prerequisites

- **Essential**
  - Solid recent experience of C...
  - Structured data types (structs and arrays)
  - Declaration and use of pointers
  - Function declaration, definition and use
  - Call by value, call by pointer
  - Dynamic memory allocation (malloc, free)
- **Beneficial**
  - Familiarity with ISO C
  - Experience working in a significant C project

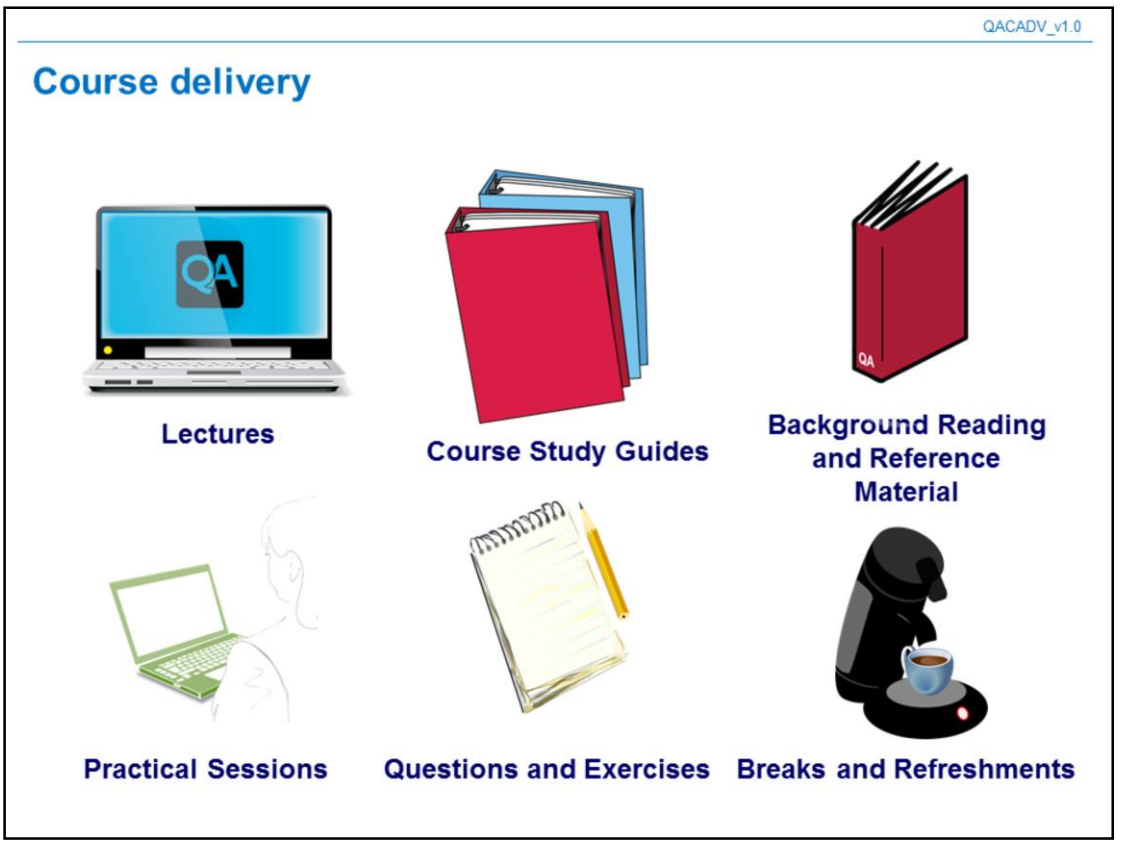
If you feel that you are on the wrong course, tell us now! We will try to help you find the right one. We do make some assumptions about you, as stated in the course outline, the course catalogue and on this slide!




Standard C and K&R C sometimes do things in slightly different ways which are, unfortunately, incompatible. These will be discussed in the course.

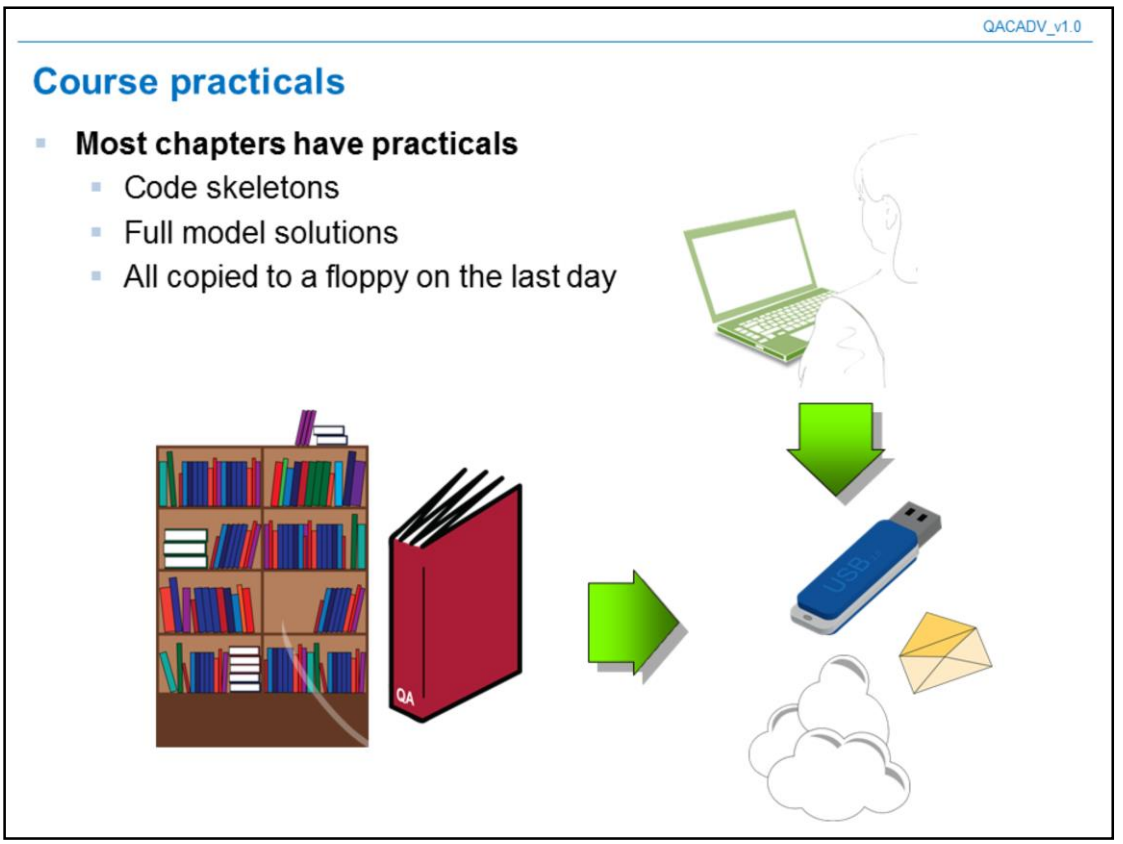
The course covers many interesting areas of the standard library include dynamic memory allocation, signal handling, and writing variadic functions (functions that take a variable number of parameters) such as `printf` and `scanf`.

Later in the course, complex data structures (such as linked lists and binary trees) will be investigated and the code required to support them will be explored.




Each chapter has well defined objectives and covers a topic in detail. The lecture notes are included in the course study guide, with supporting notes per page providing additional information and description.

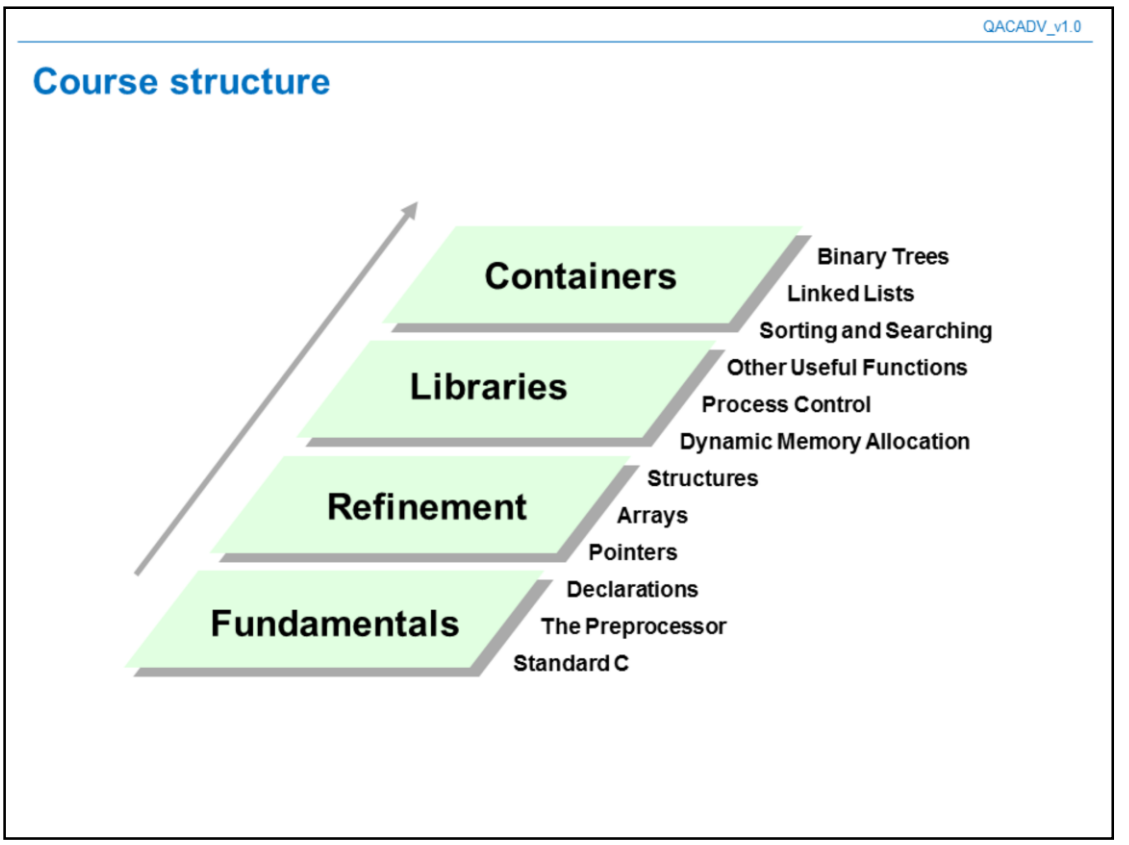
Nearly all chapters have an associated practical session. The main questions use knowledge gained in the lecture, with optional questions sometimes relying on the additional end of chapter material.




The practical questions are at the end of the chapter they refer to. This course does not focus on any vendor specific compiler/library details, preferring instead to concentrate on the ISO C Standard.

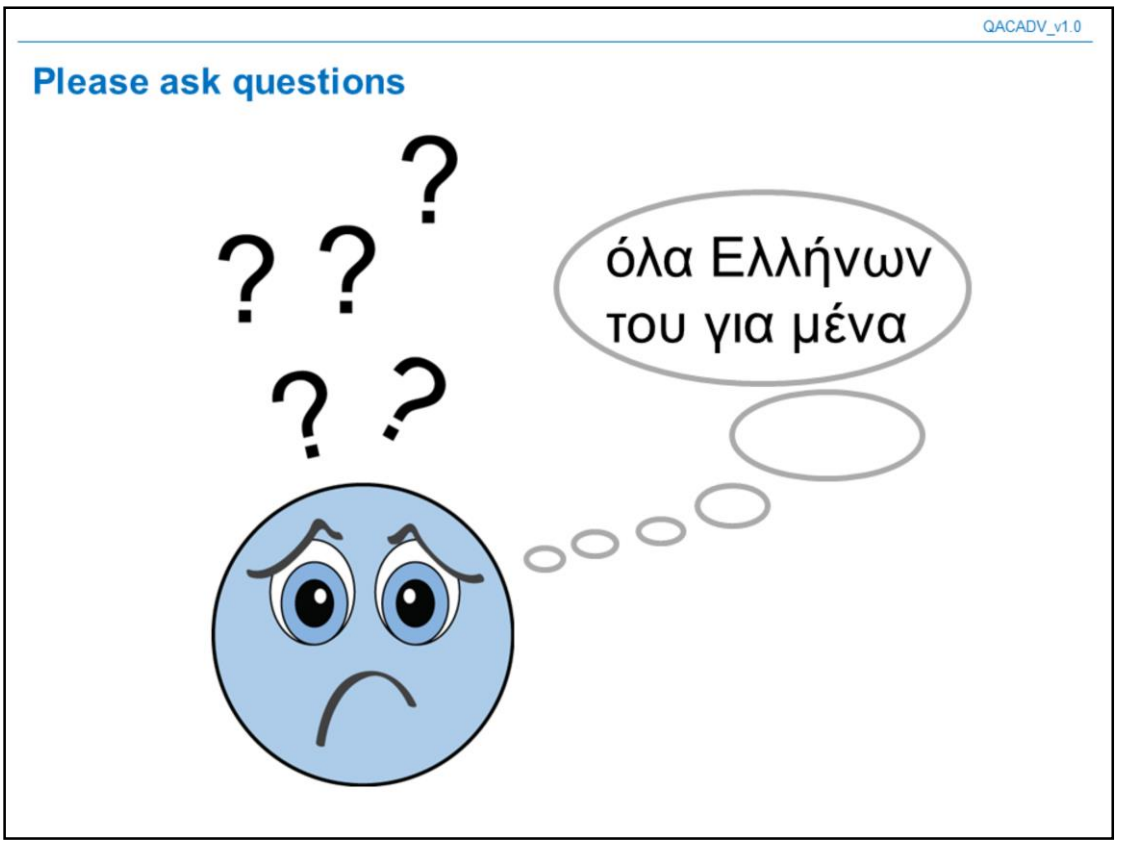
The practical questions for each chapter are arranged so that they get progressively harder. Opening questions are based on the core material of the chapter, with later optional questions being based on more advanced material or detail.

At the end of the course you can take away a copy of all the practicals, and your worked solutions.




This course has been designed as a set of progressive sections, each of which builds on the section before.

The lecturer may choose to reorder certain chapters or omit less critical ones depending on the amount of time available.




Please feel free to ask any questions at any time!

The importance of asking questions cannot be overstated. Do not be afraid to ask questions. Later chapters build on earlier chapters, so it is especially important to ask questions early on. There is no such thing as a stupid question.

Chinese proverb: "He who asks is a fool for five minutes, but he who does not ask remains a fool forever."

"To learn we must be willing to make mistakes." (Gerald Weinberg; The Psychology of Computer Programming, Dorset House, 0-932-633420)

Just in case you need a hint. όλα Ελλήνων του για μένα really means "It's all Greek to me" in Greek, naturally ☺