MAN T&EC

The Manchester and North HPC
Training and Education Centre

Fortran 90

A conversion course for Fortran 77 programmers

1 Overview

The Fortran 90 standard represents a major development in the language, allowing new programming constructs and the ability to express the data-parallel nature of a problem.

This course covers the transition from the programming language Fortran 77 to the more modern Fortran 90. The course was developed by the Manchester and North High Performance Computing Training & Education Centre (*MAN T&EC*) using the Manchester Computing Centre Fortran 90 course as a basis. This course is of 3 days duration, involving hands on sessions. Without hands on sessions, the course is of one day duration.

2 Target Audience

The course is aimed at experienced Fortran 77 programmers who require an understanding of the principles and new features of Fortran 90. It may also be suitable for programmers familiar with languages such as C or Pascal, but not for complete beginners in programming.

3 Skills Required

- Experience of Fortran 77 programming or other high level languages
- Familiarity with editor basics

4 Skills Gained

- The ability to develop, compile and execute Fortran 90 programs on at least one platform
- Knowledge of features new to Fortran 90:
    - New source form and derived types
    - Control structure
    - Procedures and modules
    - Array processing
    - Dynamic memory allocation
    - Pointers

  - New input/output features

5 Material Provided

- Course introduction, syllabus and overview
- Student notes
- Overhead masters
- Speaker notes for overheads
- Resource list
- Source code of example programs and solutions to exercises
- Feedback forms

6 Course Syllabus

6.1 Introduction

- History
- Objectives of Fortran 90
- New features summary
- Availability of compilers
- References
- Resource list
- Fortran 90 at MCC
- Recommended Coding Style

6.2 Sources, Types and Control Structures

- Free source format and character set
- Specifications
- IMPLICIT NONE statement
- KIND concept
- Derived types
- Control structures
  - IF
  - DO loops
  - CASE construct

6.3 Procedures and Modules

- Concept of program units (main program, internal, external and module procedures, modules)
- Concept of explicit and implicit interfaces and INTERFACE blocks
- Internal procedure and CONTAINS statement

- Procedure Arguments
  - Optional arguments
  - Keyword arguments
  - INTENT statement
  - Procedures as arguments
- RESULT clause for functions
- Array-valued functions
- Recursive procedures
- Generic procedures
- Operator and assignment overloading
- Modules
  - Data and/or procedures sharing
  - USE statement
  - PRIV ATE/PUBLIC objects
- Concept of scope
- Program structure
  - Order of statements
  - Program unit forms
  - Summary

6.4 Array Processing

- Terminology
- Specifications
- Whole array operations
- WHERE statement and construct
- Elemental intrinsic procedures
- Array sections
- Element renumbering
- Zero-sized arrays
- Array assignment
- Array constructors
- Assumed shape arrays
- Automatic arrays
- Dynamic (allocatable) arrays
- Array intrinsic procedures

6.5 Pointers and Dynamic Data Structures

- Concept of pointers
- Specifications
- Pointer assignment
- Pointer status

- Dynamic storage
- Pointer arguments
- Pointer functions
- Arrays of pointer
- Linked list

6.6 New Input/Output Features

- Non advancing I/O
- INQUIRE by I/O list
- NAMELIST
- New edit descriptors
- New statement specifiers

6.7 Intrinsic Procedures

- Elemental procedures
- Inquiry functions
- Transformational functions
- Nonelemental subroutines

6.8 Redundant Features

Description of redundant features in Fortran 77 and corresponding new features in Fortran 90

6.9 Further Development

Corrections/new features/HPF/MPI