

**Course Name:** Advanced RPG4 Programming & ILE

**Course Code:** ISR4A

**Duration:** 4 Days

**Price:** £ 2,300 plus VAT

**Prerequisites:** A basic understanding of RPG programming techniques as covered in the RPG4 Programming course (ISR4P).

**Course Outline:**

Review of RPGIV Keywords, Operation Codes & Built In Functions

Free format Control, File and Definition statements

Using Date & Time Variables and Functions

Using Null Capable fields

Compile Time and Run Time arrays

Data Structures [Basic, Multiple Occurrence & Multi-dimensional]

Data Areas

Pointers & Dynamic Storage

Introduction to teraspace storage model

Commitment Control & Journalling Concepts

Sub-procedures

Local & Global Variables & Procedures (Export)

ILE Concepts

Debug Considerations & Encryption

Module & Service Program Creation & Management

Binding Directories & Binder Source

Activation Group Considerations

**Course Content:**

**RPGIV review and Update**

- Be aware of all RPGIV operation codes
- Understand the commonly used built in functions
- Be clear on the course objectives & pre-requisites
- Be able to code Free form Control statements CTL-
- Be aware of all Free form definition statements DCL-  
Files, Variables, Procedures

## Date & Time variables plus null capable fields

- Define initialise and use variable types:  
Date, time, timestamp
- Covert non-date variables to date variables  
%DATE , %TIME, %TIMESTAMP
- Manipulate data in true date variables  
%DAYS, %MONTHS, %YEARS, %DIFF, %SUBDT  
(%HOURS, %MINUTES, %SECONDS, %MSECONDS)
- Validate the content of a date variable (TEST)
- Convert true date variables to non-date variables  
%CHAR, %DEC
- Define & Use null capable fields:  
ALWNULL, %NULLIND

## Arrays

- Declare and use an Array
- Differentiate between compile & run time arrays
- Provide compile time data
- Use the repeat operation code FOR
- Understand the legacy code DO
- Understand how to use the array BIF's:  
%XFOOT, %LOOKUPxx, %SUBARR
- Be aware of legacy op codes: XFOOT, LOKUP
- Sort array content: SORTA

## Data Structures

- Define and use a data structure
- Recognise different types and uses of data structures
- Hard code the sub structure of a data structure
- Understand how file information data structures can be used in report programming
- Redefine the sub-structure – OVERLAY
- Duplicate structures with QUALIFIED

- Initialise data structure and sub-structure
- Code and use a program status data structure
- Code and use a file information data structure
- Code free format data structure statements

### **External and Multiple Definitions**

- Retrieve a sub-structure from an external file EXTNAME
- Use other structures: LIKERECD, LIKEDS, QUALIFIED
- Rename sub-structure fields EXTFLD, PREFIX
- Use long fields names from external definition ALIAS
- Code and use a multiple occurrence data structure
- Overlay arrays within a data structure
- Use array BIF's %ELEM, %SIZE, %OCCUR
- Understand and replace MOVEA legacy op code
- Sort linked arrays
- Read multi-dimensional arrays & Complex Data structures
- Sort and Search techniques of Data Structure Arrays

### **Data Areas**

- Define and use a data area
- Describe and use the LDA
- Understand and use the operation codes:  
    IN, OUT, UNLOCK
- Link Data areas to Data Structures
- Use and declare a user controlled data area
- Free format Data Area Definition Statements
- Soft Code Data Areas

### **Pointers and Dynamic Storage**

- Define and use a pointer variable
- Define storage space at run-time (based)
- Use Dynamic Storage
- Understand and use Dynamic storage commands:  
    %ADDR, %ALLOC, %REALLOC & DEALLOC
- Declare Variable length & Null terminated strings

## Commitment Control

- Understand commitment control requirements
- Understand how Commitment Control works
- Understand when to use commitment control
- Maintain files under commitment control
- Include commitment control into RPG program
- Use a MONITOR error trap

## Sub-procedures and Optional Parameters

- Code a Sub-procedure
- Understand local & global variables
- Understand the need for and code a prototype
- Use the operation code CALLP
- Alternative parameter passing methods:
  - By reference, Read-only Reference, Value
- Define and use optional parameters:
  - \*OMIT, \*NOPASS, %PARMS
- Default Activation Group considerations
- Faster Return Values %PARMNUM, RTNPARM
- Code a reusable sub-procedure

## ILE Concepts and Introduction

- ILE v OPM
- Relationship between program and module objects
- Dynamic call v Static call
- ILE program and module object information – DSPPGM, DSPMOD
- Convert CLP to CLLE & modules. CL procedure Call - CALLPRC
- Binding CL and RPG together into a single program
- Update ILE program objects
- Include copyright into module objects
- Application isolation with Activation groups
- Controlling which activation group is used
- Reclaim resources for an activation group
- Bindable API's

## RPG Modules

- Bind RPG modules into programs
- Make source code available for interactive debug
- Define active PEP
- Understand export / import concepts
- Using export on sub-procedure definitions
- Define and use NOMAIN modules
- Code Cycle-Main and Linear-Main procedures
- Teraspace storage model v Single-level storage model

## Service Programs & Binding Directories

- Bind by copy v Bind by reference
- Service Program creation and Maintenance
- Service Program exports and Public Interface
- Control Service Program Signatures
- Binder Language and QSRVSR
- Public and Private procedures
- Binding Directory objects
- Use of Binding Directory to simplify the program creation process

**Follow on Courses:** Potentially the Power RPG with SQL course (ISSQE) and the Introduction to Subfile Programming (ISRPS) course. Also consider the Rational Developer for i course (ISRDI).

## Schedule:

2026

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec

Dates for this course will be scheduled on demand so please email or call for further details.