JACKSON STRUCTURED PROGRAMMING

This course provides a thorough and detailed set of procedures for writing good structured code based on a data-driven design method. The emphasis is on producing efficient, logically sound and easy to maintain program code. However, no programming language is taught.

Audience

The course is intended for trainee, junior and experienced programmers who need to be able to write programs in a clear and well-structured manner.

Prerequisites

An aptitude for programming would be useful but no prior programming experience or program design is necessary. The delegate, however, should be open-minded when it comes to thinking about program design. Following this course, the delegate may then attend one of the programming courses, such as **COBOL Programming**

Duration

Three days (but may be extended if more time required). The course is taught using desk exercises in each topic to enable the delegate to master the techniques. There are no computer-based practical exercises.

Course objectives

On completion of this course the delegate will be able to:

- understand what is system development
- understand the need for good program design
- produce file data structures
- merge data structures
- create basic program structures
- define program operations
- decide upon program conditions
- create a full program structure
- produce program pseudo-code
- produce complex designs
- have an understanding of advanced design techniques

Course contents

Introduction to Program Development

The delegate is given an overview of program application development with the emphasis on the program specification, and concludes with the history of design methods.

Introduction to JSP

The delegate is introduced to the basics of the method, the five steps to deriving an efficient program, and the basic constructs of a program, namely: Sequence, Selection and Iteration. All this is demonstrated using an example of the method.

Creating Data Structures

The first part of the design method is discussed more fully with the emphasis on designing the data structures both physically and logically. Using the three basic constructs creates a data structure for each file.

Creating the basic Program Structure

After all the data structures have been drawn, they must be merged into a single structure. This is done by identifying correspondences (a *one-to-one* relationship) between the entities of each data structure.

Providing Functionality to the Program Structure

In this stage of the method, the actions (*operations*) that the program is to perform must be listed and assigned to the basic program structure. The conditions the program needs to enable correct logical processing of the data will also need to be discussed and assigned.

JACKSON STRUCTURED PROGRAMMING

Course contents continued

Schematic Logic

Although the previous stage produces a complete program structure, it often the case that schematic logic (*pseudo-code*) is created to enable the programmer to code more easily in the chosen computer language. The terminology and the layout of JSP pseudo-code are discussed.

Adapting the Pseudo-Code

The delegate is shown how to enhance the pseudo-code to take advantage of the native computer language's constructs. Examples using popular high-level languages are given.

Introducing Advanced Design Techniques

The delegate is introduced briefly to techniques such as *Validation*, *Multiple read-ahead* and *Backtracking*, *Two-file Merge* and *Collation*, and *Structure Clashes*.