PGA-101: Programming in ADA

Course Details

Course Code: PGA-101

Duration: 5 days

Notes:

- This course syllabus should be used to determine whether the course is appropriate for the students, based on their current skills and technical training needs.
- Course content, prices, and availability are subject to change without notice.
- Terms and Conditions apply

Elements of this syllabus are subject to change.

About this course

Ada is a powerful and extensive programming language well suited for large projects. It has safety features which designed to minimise the occurrence of common programming errors, allowing complex software to be developed with more confidence. It is used extensively by the military and defence organisations, and other large organisations such as those in the banking sector.

At Course Completion

After completing this course, students will be able to:

- Understand the design goals of the language
- Write and understand moderately complex Ada programs
- Select and design appropriate data types
- Utilise the Ada tasking features
- Write and use generic Ada units

Prerequisites

The student should have some programming experience.

Academy IT Pty Ltd
Harmer House
Level 2, 5 Leigh Street
ADELAIDE 5000

Email: sales@academyit.com.au
Web: www.academyit.com.au

Phone: 08 7324 9800
Brian: 0400 112 083

Table of Contents

1. Introduction

2. Overview of Ada
- Text of an Ada Program
- Notation for Ada Syntax
- Overview of the Ada Language

3. Elementary Ada Programming
- Structure of a simple program
- Object Declarations
- Enumeration Type Declarations
- Statements
- Using Separate Compilation
- Predefined Facilities

4. Data Types
- Type Declarations
- Introduction to User Defined Types
- The Form of a Type Declaration
- Integer Type Declarations
- Discrete Types
- Floating Point Type Declarations
- Fixed Point Type Declarations
- Named Numbers and Universal Expressions
- Array Type Declarations
- Record Type Declarations
- Discriminated Records
- Complex Numbers
- More about Enumeration Types
- Evaluation of Expressions in Declarations
- One of a kind arrays
- Subtypes vs Distinct Types
- Derived Types
- Exercises

5. Expressions
- Expressions
- Elementary Expressions
- Compound Expressions
- Subprograms

6. Access Types and Packages
- Access Types
- Aliased Objects
- Dynamically Allocated Objects
- Access to Objects Declarations
- Access to subprogram declarations
- Recursive Data Structures
- Packages
- Placement of Packages
- Child Packages
- Example Package Fractions
- Use Clauses
- Renames
- Packages without bodies
- An Abstract State Machine Example

7. Private Types and Generics
- Private and Limited Private Types
- The need for Private Types
- Declaration of Private Types
- Children Packages with Private Parts
- Deferred Constants
- Programmer Defined Equality
- Limited Types
- Returning Objects By Reference
- Generic Units
- Templates and Instances
- Generic Formal Parameters
- Generic Child Units and Children of Generic Units
- Defaults for Generic Parameters

8. Tasking
- Introduction to Tasks
- Reasons for Concurrent Programming
- Example: A simple example in concurrent programming
- Simple time diagrams
- Example: A Simple controller
- Example: Processing several files
- Concurrent Processes
- Familiar examples of concurrent programming
- Task Types
- Task Objects
- Task activation
- Shared Objects
- Task Communications
- Elementary Rendezvous
- Entries
- The Accept Statement
- Entry Calls: Putting it All Together

9. Task Communication
- Basic Data Protection
- Elementary Protected Objects
- Protected Operations
- Call-backs and Protected Types
- Advanced Forms of the Rendezvous
- The Select Statement

- The Type Time and the Package Ada.Calendar
- The Delay Statement
- Timed Select Alternatives
- The Terminate Alternatives
- Conditional and Timed Entry Call
- Aborting Actions of a Task
- The abort statement
- Requeuing
- Entry Families
- Task identification
- Controlling Real-Time Behaviour
- Priorities

10. Exceptions and Classwide Types

Exceptions

- Handling Exceptions
- Propagation of Exceptions
- Predefined Exceptions
- User Defined Exceptions
- Exception Occurences
- Other Topics on Exceptions
- Classwide Programming
- Object Oriented Concepts
- Derived Types
- Type Extension
- Classwide Types
- Controlled Types