Patterns Overview

An Introduction to Pattern Concepts

Non-trivial systems include many recurring design problems whose solutions are commonly repeated from place to place, and from programmer to programmer. The essence and basic structure of a solution may be repeated many times, even though the realisation is different in each case. Patterns offer a technique for capturing design and architecture, presenting and communicating architectural knowledge at all levels of a system, allowing experience to be understood and distilled. Patterns allow developers to work on and understand designs, and are not a basis for automation of design; frameworks and libraries present code-level reuse often built on common patterns.

The *Patterns Overview* course introduces basic pattern concepts and benefits, including some concrete examples. It develops introductory understanding through lectures and discussion.

Objectives

- Understand what does and does not go to make up a pattern
- Understand the beneficial role of patterns in all aspects of development
- Appreciate patterns from the strategic level down to idiomatic examples in C++ and Java

Audience

The course is suitable for software developers familiar with object-oriented principles and practices. Object-oriented programming experience is assumed, and familiarity with UML is beneficial.

Content

Design and Architecture Design · Architecture · Describing or defining an architecture · Reuse of knowledge · Patterns and pattern form · Catalogues and pattern languages

Gang of Four Patterns The "Gang of Four" design patterns catalogue · The Composite pattern · The Factory Method pattern · The Objects for States (State) pattern · The Command pattern

POSA Patterns The "Pattern-Oriented Software Architecture" system of patterns · Architectural patterns, design patterns and idioms · The Proxy pattern · The Layers pattern · The Command Processor pattern

Patterns from Other Sources The Manager pattern · The Null Object pattern · The Collections for States pattern · The Role Decoupling pattern

Idioms The Smart Pointer idiom for C++ · The Immutable Value idiom in Java · The Combined Operation idiom for distributed programming

Additional Details

Duration 1 day

Setup Projection facilities for a laptop · Whiteboards and/or flip charts

Contact Kevlin Henney · kevlin@curbralan.com · Curbralan Limited · +44 117 942 2990