incusdata (pty) ltd. 1995/04149/07
Coding · Matters
PO Box 94
Raslouw, 0109
tel : +27 12 666 2020
fax: +27 12 666 2021
www.incusdata.com

Price: R9,800.00 excl. VAT
Duration: 4.5 days
Code: OOADN

Object-Oriented Analysis & Design using UML

Description

Object Orientation, or OO, is a methodology that can be used during the whole software development life cycle: from analysis of users' needs, to design and then coding and testing. The Unified Modelling Language (UML) is an international set of diagrams for modelling your system using an object-oriented approach.
This course will teach you how to analyse, design and document your system using established OO principles. It will help you to understand the problem and the solution better, communicate it more effectively and guide you to write better code.

Objectives

After you have completed the Object Oriented Analysis and Design using UML course, you will be able to:

- Understand object-oriented concepts and principles, and the OO project lifecycle.
- Write effective use cases.
- Take part in a design session that uses Class, Responsibilities, and Collaboration (CRC) cards.
- Design a system using classes and class relationships.
- Develop activity and class diagrams.
- Develop sequence and communication diagrams.

Intended Audience

You should attend the Object Oriented Analysis and Design using UML course if:

- You are a programmer moving to an object-oriented language, like Java.
- You are a programmer using an object-oriented language, but want to understand and use object-oriented principles more effectively.
- You are a system analyst or designer, and you need to design object-oriented systems.
- You are a system architect, and you need to understand how to design and create object-oriented systems.

Prerequisites

Before you attend the Object Oriented Analysis and Design using UML course:

- You should have some experience in programming, ideally in an object-oriented language like Java, C# or C++.

If you are a business analyst and have little or no programming experience, then you should attend the Object-Oriented Analysis course which runs as the first 3 days of this course.

Course Contents

Introduction

- The evolution of the object-oriented paradigm.
- OOP compared to other programming paradigms.
- Advantages and disadvantages of OOP.

Object-Oriented Concepts and Terminology

- Classes and objects.

PO Box 94
Raslouw, 0109
tel : +27 12 666 2020
fax: +27 12 666 2021
www.incusdata.com

- Attributes and behaviours.
- Data abstraction and encapsulation.
- Polymorphism.
- Inheritance and code reuse.
- Associations and relationships between classes.

Unified Modelling Language

- History and evolution of the UML.
- UML diagrams: use case, class, object, sequence, communication, state, activity, component, package, timing, subsystem, model, deployment diagrams.
- Common extension mechanisms.
- UML modelling tools.

Object-Oriented Methodologies

- Traditional Software Development Lifecycle.
- Iterative and incremental development.
- The need for an OOAD process.
- The Rational Unified Approach (RUP).
- The Iconix method.
- Extreme Programming and Agile Modelling.

Object-Oriented Analysis

- Behaviour analysis and use cases.
- Activity diagrams.
- CRC cards.
- Domain modelling.
- Class identification and domain classes.
- UML diagrams for analysis.

Object-Oriented Design

- Responsibility driven design.
- Class design and detailed class diagrams.
- Robustness diagrams and the MVC architecture.
- Sequence, communication and state diagrams.
- Timing diagrams.
- UML diagrams for design.

Design Principles & Patterns

- What makes a design feel "right".
- Advanced design principles.
- Design pattern concepts.
- Examples of commonly used patterns.

******* *The lecturer reserves the right to modify the contents of the course to suit the needs of the delegates.*