Java Intermediate - Object Oriented Programming

Java Training

About The Course

During this 3-day course students will learn about object oriented design and understand how to start a Java based project from scratch. You will learn about the Java environment and its inner workings so you can run the code you have written in the most appropriate environment. You will also learn to write code that utilises inheritance to save unnecessary duplication, polymorphism to make your code extensible, encapsulation to keep code simple, and interfaces to allow ‘plug-and-play’ style coding. At the end of this course students should feel confident to write object oriented code that is efficient, well designed and future proof.

Duration: 3 days

Class size: 10 students max

Times: 9:00am - 5:00pm

Price: *Refer to our website for current course and package pricing*

After the course?

Each student will receive:

- Certificate of completion
- Training manual
- 12 months FREE email support
- FREE class re-sit (if necessary)

Who Should Do This Course?

This course is suitable for participants who want to extend their skills beyond the syntax of Java language and be able to write code to professional standards using best practices. This course is also relevant to people who want to learn about Object-oriented programming and principles.

Prerequisites

Participants should have completed the Java for Beginners course or have a good understanding of Java syntax and coding structure and also have a good familiarity with using IDE’s, preferably Netbeans or Ecllipse.

Content

Unit 1: Starting Out

- Understanding this Course
- Checking your Understanding of Code
- Understanding Object Oriented Programming
- Understanding Platform Independence
- Understanding Java’s Origins
- Exploring how the Internet Saved Java
- Understanding Java’s Current Place in the Internet of Things
- Exploring the Java Timeline

Unit 2: Understanding Platform Independence

- Gaining a Better Understanding of Platform Independence.
- Understanding the Compiler
- Understanding Bytecode
- Understanding Machine Code
- Understanding the Java Virtual Machine (JVM)
- Putting it All Together
- Investigating the Consequences of Platform Independence
- Looking at Languages that are Platform Independent
- Looking at Languages that are Not

- Understanding the Drawbacks of Platform Independence
- Understanding Micro, Standard and Enterprise Java
- Understanding the Importance of Specifications
- Joining the Java Community Process
- Understanding the Java Environment
- Understanding the Function of A JVM
- Understanding how the JVM and Specification Work Together
- Understanding why Some JVMs Differ from the Specification
- Understanding the Java Developer's Kit (JDK)
- Finding the JDK
- Downloading the JDK
- Exploring the Contents of the JDK
- Running Java
- Understanding how Java Runs
- Compiling your Code
- Running your Code
- Looking at the Results

Unit 3: Understanding Object Oriented Programming

- Getting a Better Understanding of Object Oriented Programming
- Looking at Languages that are Object Oriented
- Looking at Languages that are Not Object Oriented
- Understanding why Some People Say It's Easy and Others Say It's Hard
- Comparing Object Oriented Code with Procedural Code
- Analysing an Object Using a Simple Example
- Looking at Different Types of Pens
- Looking at Objects that are Not Pens
- Checking Accuracy
- Considering the Domain of Our Problem
- Understanding Objects - Fields Verses Methods
- Understanding Objects - Object Verses Class
- Understanding Objects - Other Principles
- Coding Objects in Java
- Coding Blocks
- Coding Classes
- Coding Methods
- Coding Fields
- Coding Packages

Unit 4: Gaining a Deeper Understanding of Classes

- Understanding Memory: Stack Verses Heap
- Understanding the difference between a primitive and an object
- Investigating AutoBoxing and AutoUnboxing
- Understanding the Class Loader
- Understanding the Garbage Collector
- Understanding How Java Passes Parameters

Unit 5: Understanding Inheritance and Polymorphism

- Looking at an Example of Inheritance
- Reviewing Our Pen's and Pencil Example
- Coming to Terms with the Concept of Inheritance
- Understanding When to Use Inheritance
- Sharing Features
- Understanding the Advantages
- Coding Inheritance in Java
- Coding the Writing Implement
- Coding the Pen
- Coding the Pencil
- Coding a Permanent Marker
- Understanding the Concept of Single Inheritance
- Understanding Why We Have Single Inheritance.
- Coding Constructors to Utilise Inheritance.
- Understanding the Inheritance of Constructors
- Creating Multiple Constructors
- Chaining Constructors
- Explicitly Declaring the Default Constructor
- Calling the Parent Constructor
- Understanding what Polymorphism Is
- Casting your Object
- Understanding Casting
- Using Instance Of
- Writing Code Using Polymorphism
- Using Polymorphism when Passing Parameters
- Looking Deeper at Generic Code Using Polymorphism
- Understanding Virtual Method Invocation
- Investigating the Advantages of Generic Code
- Understanding Polymorphism Through an Everyday Example

Unit 6: Writing Code Using Encapsulation (Optional)

- Encapsulating a Class
- Understanding the Concept of Encapsulation
- Writing Getters and Setters
- Looking at Some Examples
- Making Fields Private
- Looking at a Complete Example
- Making an Object Immutable
- Understanding Immutable Objects
- Looking at the Advantages
- Looking at a Class Before Immutability
- Creating Constructor
- Removing Setters
- Looking at the Class After Immutability

- Investigation Encapsulation and Immutability with NetBeans
- Generating Getters and Setters
- Making Fields Private
- Creating Constructors
- Using Encapsulation in Real Life Examples
- Adding Validation
- Adding Formatting
- Writing Code to Interpret Input
- Writing Code for Data Independence
- Looking at the Finished Product

Unit 7: Using the Abstract, Final, and Static Keywords

- Using Access Modifiers
- Declaring and Using Final Variables
- Declaring and Using Final Classes
- Declaring and Using Final Methods
- Understanding Static Fields
- Declaring and Using Static Fields
- Creating a Static Field
- Comparing the Static and Final Keywords
- Understanding Static Methods
- Declaring and Using Static Methods
- Understanding the Limitations of Static Methods
- Understanding Static Classes
- Declaring and Using Static Classes
- Creating a Static Class
- Declaring and Using Static Blocks
- Declaring and Using Helper Classes
- Using The Static Import Command
- Understanding the Meaning of the Word ‘Abstract’
- Declaring and Using Abstract Classes
- Understanding Abstract Classes
- Declaring an Abstract Class
- Using an Abstract Class
- Declaring and Using Abstract Methods
- Understanding Abstract Methods
- Declaring an Abstract Method
- Using an Abstract Method
- The Relationship Between Abstract Methods and Abstract Classes.

Unit 8: Writing Better Code with Enumerated Datatypes

- Using Constants to Represent Datatypes
- Understanding Why This Approach Fails
- Looking at a Better Solution
- Coding with Enumerated Datatypes
- Investigating More Complex Enumeration
- Putting it All Together

Unit 9: Developing Interfaces

- Introducing Interfaces
- Understanding the Concept of an Interface
- Understanding the Function of an Interface
- Looking at an Example
- Implementing the Interface
- Programming Bottom Up with Interfaces
- Programming Top Down with Interfaces
- Creating an Interface for Global Constants
- Creating an Interface to Serve as a Marker
- Programming to an Interface
- Writing Code That Uses an Interface
- Revisiting Polymorphism
- Using the Interface in Our Code

Unit 10: Using Interfaces

- Casting the Object from Its Interface to Its Implementation
- Changing the Implementation of Our Interface
- Investigating the Advantages of Interfaces
- Writing New Code Using Interfaces
- Using Interfaces to Understand Existing Code
- Using Interfaces to Help with Possible Future Code
- How Interfaces and Encapsulation Are Different
- Investigating Multiple Inheritance with Interfaces
- Writing Code for Multiple Inheritance
- Understanding How It Works
- Comparing Implementation and Extending
- Looking at an Examples of Inheritance Hierarchies
- Investigating Changes in Java SE
- Putting Static Methods in an Interface
- Including Default Methods in an Interface
- Understanding How Default Methods Work

Unit 11: Understanding How Lambda Expressions Work

- Relationship Between a Lambda Expression and its Interface Name
- Finding the Name of the Method
- Working Out the Parameters
- Assigning Code to the Method
- Investigating Where Lambda Expression Can Be Used
- Understanding What Is So Revolutionary About Lambdas

Unit 12: Use of Object Oriented Principles in the API

- Investigating an Example of Static Classes
- Understanding the Math Helper Class
- Using the Math Class

- Using the Static Methods and Fields in the Math Class
- Investigating an Example of Inheritance
- Understanding Inheritance in Exceptions
- Catching the Right Exceptions
- Ordering the Catch Clauses
- Raising your Own Exceptions
- Creating your Own Exceptions
- Wrapping Existing Exceptions
- Investigating an Example of Interfaces and Polymorphism
- Understanding Collection Interfaces
- Coding to the Interface
- Looking at the Advantages

Looking for course dates?

To view a full list of course dates, please visit our website at www.dynamicwebtraining.com.au
Alternatively please contact our office on 1300 888 724