

ASP.NET CORE MVC

ASP.NET CORE MVC

Develop cloud-ready web applications using Microsoft's latest framework, ASP.NET Core MVC

Prerequisites / Further Training

- [Introduction to Programming](#)
- [Beginner C#.NET](#) (C# Fundamentals and Object – Orientation)
- [Advanced C#.NET](#) (Building practical Apps)
- [SQL Querying](#) (Database design and querying)
- *ASP.NET Core MVC (Web App Development)*
- Entity Framework

Also have a look at our [.NET Bootcamp](#) and save

Intended Audience

This course is useful for C# or ASP.NET Forms programmer who wants to learn the MVC (Model View Controller) methodology.

This course covers all the topics of the Microsoft Exam for ASP.NET MVC. After completing this course you should be able to prepare for the Microsoft exam and understand all the concepts to pass quickly, just by doing a few mock exams.

Further Training

You may consider doing the certification exam courses:

MCPD – Web Apps

Course Material

Included in the course price.

Course Contents

Day1:

A First MVC Application

- Adding the First Controller
- Understanding Routes
- Rendering Web Pages
- Creating and Rendering a View
- Adding Dynamic Output
- Creating a Simple Data-Entry Application
- Setting the Scene
- Designing a Data Model
- Linking Action Methods
- Building the Form
- Setting the Start URL
- Handling Forms
- Adding Validation
- Styling the Content

The MVC Pattern

- The History of MVC
- Understanding the MVC Pattern
- Understanding the Domain Model
- The ASP.NET Implementation of MVC
- Comparing MVC to Other Patterns
- Building Loosely Coupled Components
- Using Dependency Injection
- Using a Dependency Injection Container
- Getting Started with Automated Testing
- Understanding Unit Testing
- Understanding Integration Testing

Essential Language Features

- Using Automatically Implemented Properties
- Using Object and Collection Initializers
- Using Extension Methods
- Applying Extension Methods to an Interface

- Creating Filtering Extension Methods
- Using Lambda Expressions
- Using Automatic Type Inference
- Using Anonymous Types
- Performing Language Integrated Queries
- Understanding Deferred LINQ Queries
- Using Async Methods
- Applying the async and await Keywords

Working with Razor

- Preparing the Example Project
- Defining the Model
- Defining the Controller
- Creating the View
- Working with the Model Object
- Working with Layouts
- Creating the Layout
- Applying a Layout
- Using a View Start File
- Shared Layouts
- Using Razor Expressions
- Inserting Data Values
- Setting Attribute Values
- Using Conditional Statements
- Enumerating Arrays and Collections
- Dealing with Namespaces

Essential Tools for MVC

- Creating the Model Classes
- Adding the Controller
- Adding the View
- Using Ninject
- Understanding the Problem
- Adding Ninject to the Visual Studio Project
- Getting Started with Ninject
- Setting up MVC Dependency Injection

- Creating Chains of Dependency
- Specifying Property and Constructor Parameter Values
- Using Conditional Binding
- Setting the Object Scope
- Unit Testing with Visual Studio
- Creating the Unit Test Project
- Creating the Unit Tests
- Running the Unit Tests (and Failing)
- Implementing the Feature
- Testing and Fixing the Code
- Using Moq
- Understanding the Problem
- Adding Moq to the Visual Studio Project
- Adding a Mock Object to a Unit Test
- Creating a More Complex Mock Object

Day2:

A Real Application

- Creating the Visual Studio Solution and Projects
- Installing the Tool Packages
- Adding References Between Projects
- Setting Up the DI Container
- Starting the Domain Model
- Creating an Abstract Repository
- Making a Mock Repository
- Displaying a List of Products
- Adding a Controller
- Adding the Layout, View Start File and View
- Setting the Default Route
- Preparing a Database
- Creating the Database
- Defining the Database Schema
- Adding Data to the Database
- Creating the Entity Framework Context
- Creating the Product Repository
- Adding Pagination

- Displaying Page Links
- Improving the URLs
- Styling the Content
- Installing the Bootstrap Package
- Applying Bootstrap Styles to the Layout
- Creating a Partial View

Navigation

- Adding Navigation Controls
- Filtering the Product List
- Refining the URL Scheme
- Building a Category Navigation Menu
- Correcting the Page Count
- Building the Shopping Cart
- Defining the Cart Entity
- Adding the Add to Cart Buttons
- Implementing the Cart Controller
- Displaying the Contents of the Cart

The Shopping Cart

- Using Model Binding
- Creating a Custom Model Binder
- Completing the Cart
- Removing Items from the Cart
- Adding the Cart Summary
- Submitting Orders
- Extending the Domain Model
- Adding the Checkout Process
- Implementing the Order Processor
- Registering the Implementation
- Completing the Cart Controller
- Displaying Validation Errors
- Displaying a Summary Page

Mobile Interface

- Putting Mobile Web Development in Context

- Doing Nothing (Or As Little As Possible)
- Using Responsive Design
- Creating a Responsive Header
- Creating a Responsive Product List
- Creating Mobile Specific Content
- Creating a Mobile Layout
- Creating the Mobile Views

Administration Section

- Adding Catalog Management
- Creating a CRUD Controller
- Creating a New Layout
- Implementing the List View
- Editing Products
- Creating New Products
- Deleting Products

Day3:

Security & Finishing Touches

- Securing the Administration Controller
- Creating a Basic Security Policy
- Applying Authorization with Filters
- Creating the Authentication Provider
- Creating the Account Controller
- Creating the View
- Image Uploads
- Extending the Database
- Enhancing the Domain Model
- Creating the Upload User Interface Elements
- Saving Images to the Database
- Implementing the GetImage Action Method
- Displaying Product Images

Overview of MVC Projects

- Working with Visual Studio MVC Projects

- Creating the Project
- Understanding MVC Conventions
- Debugging MVC Applications
- Preparing the Example Project
- Launching the Visual Studio Debugger
- Causing the Visual Studio Debugger to Break
- Using Edit and Continue
- Using Browser Link

URL Routing

- Creating the Controllers
- Creating the View
- Setting the Start URL and Testing the Application
- Introducing URL Patterns
- Creating and Registering a Simple Route
- Using the Simple Route
- Defining Default Values
- Using Static URL Segments
- Defining Custom Segment Variables
- Using Custom Variables as Action Method Parameters
- Defining Optional URL Segments
- Defining Variable-Length Routes
- Prioritizing Controllers by Namespaces
- Constraining Routes
- Constraining a Route Using a Regular Expression
- Constraining a Route to a Set of Specific Values
- Constraining a Route Using HTTP Methods
- Using Type and Value Constraints
- Defining a Custom Constraint
- Using Attribute Routing
- Enabling and Applying Attribute Routing
- Creating Routes with Segment Variables
- Applying Route Constraints
- Using a Route Prefix

Advanced Routing Features

- Simplifying the Routes
- Adding the Optimization Package
- Updating the Unit Test Project
- Generating Outgoing URLs in Views
- Using the Routing System to Generate an Outgoing URL
- Targeting Other Controllers
- Passing Extra Values
- Specifying HTML Attributes
- Generating Fully Qualified URLs in Links
- Generating URLs (and Not Links)
- Generating Outgoing URLs in Action Methods
- Generating a URL from a Specific Route
- Customizing the Routing System
- Creating a Custom RouteBase Implementation
- Creating a Custom Route Handler
- Working with Areas
- Creating an Area
- Populating an Area
- Resolving the Ambiguous Controller Issue
- Creating Areas with Attributes
- Generating Links to Actions in Areas
- Routing Requests for Disk Files
- Configuring the Application Server
- Defining Routes for Disk Files
- Bypassing the Routing System
- URL Schema Best Practices
- Make Your URLs Clean and Human-Friendly
- GET and POST : Pick the Right One

Controllers and Actions

- Introducing the Controller
- Creating a Controller with IController
- Creating a Controller by Deriving from the Controller Class
- Receiving Request Data
- Getting Data from Context Objects

- Using Action Method Parameters
- Producing Output
- Understanding Action Results
- Returning HTML by Rendering a View
- Passing Data from an Action Method to a View
- Performing Redirections
- Returning Errors and HTTP Codes

Day4:

Filters

- Using Filters
- Introducing the Filter Types
- Applying Filters to Controllers and Action Methods
- Using Authorization Filters
- Applying the Custom Authorization Filter
- Using the Built-in Authorization Filter
- Using Authentication Filters
- Understanding the IAuthenticationFilter Interface
- Implementing the Authentication Check
- Combining Authentication and Authorization Filters
- Handling the Final Challenge Request
- Using Exception Filters
- Creating an Exception Filter
- Applying the Exception Filter
- Using a View to Respond to an Exception
- Using the Built-in Exception Filter
- Using Action Filters
- Implementing the OnActionExecuting Method
- Implementing the OnActionExecuted Method
- Using Result Filters
- Using the Built-in Action and Result Filter Class
- Using Other Filter Features
- Filtering Without Attributes
- Using Global Filters
 - Ordering Filter Execution
 - Overriding Filters

Controller Extensibility

- Creating a Custom Controller Factory
- Dealing with the Fallback Controller
- Instantiating Controller Classes
- Implementing the Other Interface Methods
- Registering a Custom Controller Factory
- Working with the Built-in Controller Factory
- Prioritizing Namespaces
- Customizing `DefaultControllerFactory` Controller Instantiation
- Creating a Custom Action Invoker
- Using the Built-in Action Invoker
- Using a Custom Action Name
- Using Action Method Selection
- Improving Performance with Specialized Controllers
- Using Sessionless Controllers
- Using Asynchronous Controllers

Views

- Creating a Custom View Engine
- Creating a Custom `IView`
- Creating an `IViewEngine` Implementation
- Registering a Custom View Engine
- Testing the View Engine
- Working with the Razor Engine
- Preparing the Example Project
- Understanding Razor View Rendering
- Configuring the View Search Locations
- Adding Dynamic Content to a Razor View
- Using Layout Sections
- Using Partial Views
- Using Child Actions

Helper Methods

- Creating Custom Helper Methods

- Creating an Inline Helper Method
- Creating an External Helper Method
- Managing String Encoding in a Helper Method
- Using the Built-In Form Helper Methods
- Creating Form Elements
- Specifying the Route Used by a Form
- Using Input Helpers
- Creating Select Elements

Templated Helper Methods

- Using Templated Helper Methods
- Generating Label and Display Elements
- Using Whole-Model Templated Helpers
- Using Model Metadata
- Using Metadata to Control Editing and Visibility
- Using Metadata for Labels
- Using Metadata for Data Values
- Using Metadata to Select a Display Template
- Applying Metadata to a Buddy Class
- Working with Complex Type Properties
- Customizing the Templated View Helper System
- Creating a Custom Editor Template
- Creating a Generic Template
- Replacing the Built-in Templates

Day5:

URL and Ajax Helper Methods

- Creating Basic Links and URLs
- Using MVC Unobtrusive Ajax
- Creating the Synchronous Form View
- Preparing the Project for Unobtrusive Ajax
- Creating an Unobtrusive Ajax Form
- Preparing the Controller
- Creating the Ajax Form
- Understanding How Unobtrusive Ajax Works

- Setting Ajax Options
- Ensuring Graceful Degradation
- Providing the User with Feedback While Making an Ajax Request
- Prompting the User Before Making a Request
- Creating Ajax Links
- Ensuring Graceful Degradation for Links
- Working with Ajax Callbacks
- Working with JSON
- Adding JSON Support to the Controller
- Processing JSON in the Browser
- Preparing Data for Encoding
- Detecting Ajax Requests in the Action Method

Model Binding

- Understanding Model Binding
- Using the Default Model Binder
- Binding to Simple Types
- Binding to Complex Types
- Binding to Arrays and Collections
- Manually Invoking Model Binding
- Dealing with Binding Errors
- Customizing the Model Binding System
- Creating a Custom Value Provider
- Creating a Custom Model Binder
- Registering the Custom Model Binder

Model Validation

- Explicitly Validating a Model
- Displaying Validation Errors to the User
- Displaying Validation Messages
- Displaying Property-Level Validation Messages
- Using Alternative Validation Techniques
- Performing Validation in the Model Binder
- Specifying Validation Rules Using Metadata
- Defining Self-Validating Models

- Performing Client-Side Validation
- Enabling Client-Side Validation
- Using Client-Side Validation
- Understanding How Client-Side Validation Works
- Performing Remote Validation

Bundles

- Profiling Script and Style Sheet Loading
- Using Script and Style Bundles
- Adding the NuGet Package
- Defining the Bundles
- Applying Bundles
- Optimizing the JavaScript and CSS Files

Web API and Single-page Applications

- Understanding Single-page Applications
 - Creating the Model
 - Adding the NuGet Packages
 - Adding the Controller
 - Adding the Layout and Views
 - Setting the Start Location and Testing the Example Application
 - Using Web API
 - Creating the Web API Controller
 - Testing the API Controller
 - Understanding How the API Controller Works
 - Understanding API Controller Action Selection
 - Mapping HTTP Methods to Action Methods
 - Using Knockout for Single-page Applications
 - Adding the JavaScript Libraries to the Layout
 - Implementing the Summary
 - Implementing the Create Feature
 - Completing the Application
 - Simplify the Home Controller
 - Manage Content Visibility
-

Duration and pricing

- *Full-time* over 5 days (R9995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
 - [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course price.
 - Prices exclude Vat for Vat-registered companies

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950 including the course material and guidance session.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

ASP.NET MVC 5

Prerequisites / Further Training

- [Introduction to Programming](#)
- [Beginner C#.NET](#) (C# Fundamentals and Object – Orientation)
- [Advanced C#.NET](#) (Building practical Apps)
- [SQL Querying](#) (Database design and querying)
- *ASP.NET MVC5 (Web App Development)*
- ASP.NET Web API (More Web App Development)
- Entity Framework

Also have a look at our [.NET Bootcamp](#) and save

Intended Audience

This course is useful for C# or ASP.NET Forms programmer who wants to learn the MVC (Model View Controller) methodology.

This course covers all the topics of the Microsoft Exam for ASP.NET MVC 5. After completing this course you should be able to prepare for the Microsoft exam and understand all the concepts to pass quickly, just by doing a few mock exams.

Further Training

You may consider doing the certification exam courses:

MCPD – Web Apps

Course Material

Included in the course price.

Course Contents

Day1:

A First MVC Application

- Adding the First Controller
- Understanding Routes
- Rendering Web Pages
- Creating and Rendering a View
- Adding Dynamic Output
- Creating a Simple Data-Entry Application
- Setting the Scene
- Designing a Data Model
- Linking Action Methods
- Building the Form
- Setting the Start URL
- Handling Forms
- Adding Validation
- Styling the Content

The MVC Pattern

- The History of MVC
- Understanding the MVC Pattern
- Understanding the Domain Model
- The ASP.NET Implementation of MVC
- Comparing MVC to Other Patterns
- Building Loosely Coupled Components
- Using Dependency Injection
- Using a Dependency Injection Container
- Getting Started with Automated Testing
- Understanding Unit Testing
- Understanding Integration Testing

Essential Language Features

- Using Automatically Implemented Properties
- Using Object and Collection Initializers
- Using Extension Methods
- Applying Extension Methods to an Interface

- Creating Filtering Extension Methods
- Using Lambda Expressions
- Using Automatic Type Inference
- Using Anonymous Types
- Performing Language Integrated Queries
- Understanding Deferred LINQ Queries
- Using Async Methods
- Applying the async and await Keywords

Working with Razor

- Preparing the Example Project
- Defining the Model
- Defining the Controller
- Creating the View
- Working with the Model Object
- Working with Layouts
- Creating the Layout
- Applying a Layout
- Using a View Start File
- Shared Layouts
- Using Razor Expressions
- Inserting Data Values
- Setting Attribute Values
- Using Conditional Statements
- Enumerating Arrays and Collections
- Dealing with Namespaces

Essential Tools for MVC

- Creating the Model Classes
- Adding the Controller
- Adding the View
- Using Ninject
- Understanding the Problem
- Adding Ninject to the Visual Studio Project
- Getting Started with Ninject
- Setting up MVC Dependency Injection

- Creating Chains of Dependency
- Specifying Property and Constructor Parameter Values
- Using Conditional Binding
- Setting the Object Scope
- Unit Testing with Visual Studio
- Creating the Unit Test Project
- Creating the Unit Tests
- Running the Unit Tests (and Failing)
- Implementing the Feature
- Testing and Fixing the Code
- Using Moq
- Understanding the Problem
- Adding Moq to the Visual Studio Project
- Adding a Mock Object to a Unit Test
- Creating a More Complex Mock Object

Day2:

A Real Application

- Creating the Visual Studio Solution and Projects
- Installing the Tool Packages
- Adding References Between Projects
- Setting Up the DI Container
- Starting the Domain Model
- Creating an Abstract Repository
- Making a Mock Repository
- Displaying a List of Products
- Adding a Controller
- Adding the Layout, View Start File and View
- Setting the Default Route
- Preparing a Database
- Creating the Database
- Defining the Database Schema
- Adding Data to the Database
- Creating the Entity Framework Context
- Creating the Product Repository
- Adding Pagination

- Displaying Page Links
- Improving the URLs
- Styling the Content
- Installing the Bootstrap Package
- Applying Bootstrap Styles to the Layout
- Creating a Partial View

Navigation

- Adding Navigation Controls
- Filtering the Product List
- Refining the URL Scheme
- Building a Category Navigation Menu
- Correcting the Page Count
- Building the Shopping Cart
- Defining the Cart Entity
- Adding the Add to Cart Buttons
- Implementing the Cart Controller
- Displaying the Contents of the Cart

The Shopping Cart

- Using Model Binding
- Creating a Custom Model Binder
- Completing the Cart
- Removing Items from the Cart
- Adding the Cart Summary
- Submitting Orders
- Extending the Domain Model
- Adding the Checkout Process
- Implementing the Order Processor
- Registering the Implementation
- Completing the Cart Controller
- Displaying Validation Errors
- Displaying a Summary Page

Mobile Interface

- Putting Mobile Web Development in Context

- Doing Nothing (Or As Little As Possible)
- Using Responsive Design
- Creating a Responsive Header
- Creating a Responsive Product List
- Creating Mobile Specific Content
- Creating a Mobile Layout
- Creating the Mobile Views

Administration Section

- Adding Catalog Management
- Creating a CRUD Controller
- Creating a New Layout
- Implementing the List View
- Editing Products
- Creating New Products
- Deleting Products

Day3:

Security & Finishing Touches

- Securing the Administration Controller
- Creating a Basic Security Policy
- Applying Authorization with Filters
- Creating the Authentication Provider
- Creating the Account Controller
- Creating the View
- Image Uploads
- Extending the Database
- Enhancing the Domain Model
- Creating the Upload User Interface Elements
- Saving Images to the Database
- Implementing the GetImage Action Method
- Displaying Product Images

Overview of MVC Projects

- Working with Visual Studio MVC Projects

- Creating the Project
- Understanding MVC Conventions
- Debugging MVC Applications
- Preparing the Example Project
- Launching the Visual Studio Debugger
- Causing the Visual Studio Debugger to Break
- Using Edit and Continue
- Using Browser Link

URL Routing

- Creating the Controllers
- Creating the View
- Setting the Start URL and Testing the Application
- Introducing URL Patterns
- Creating and Registering a Simple Route
- Using the Simple Route
- Defining Default Values
- Using Static URL Segments
- Defining Custom Segment Variables
- Using Custom Variables as Action Method Parameters
- Defining Optional URL Segments
- Defining Variable-Length Routes
- Prioritizing Controllers by Namespaces
- Constraining Routes
- Constraining a Route Using a Regular Expression
- Constraining a Route to a Set of Specific Values
- Constraining a Route Using HTTP Methods
- Using Type and Value Constraints
- Defining a Custom Constraint
- Using Attribute Routing
- Enabling and Applying Attribute Routing
- Creating Routes with Segment Variables
- Applying Route Constraints
- Using a Route Prefix

Advanced Routing Features

- Simplifying the Routes
- Adding the Optimization Package
- Updating the Unit Test Project
- Generating Outgoing URLs in Views
- Using the Routing System to Generate an Outgoing URL
- Targeting Other Controllers
- Passing Extra Values
- Specifying HTML Attributes
- Generating Fully Qualified URLs in Links
- Generating URLs (and Not Links)
- Generating Outgoing URLs in Action Methods
- Generating a URL from a Specific Route
- Customizing the Routing System
- Creating a Custom RouteBase Implementation
- Creating a Custom Route Handler
- Working with Areas
- Creating an Area
- Populating an Area
- Resolving the Ambiguous Controller Issue
- Creating Areas with Attributes
- Generating Links to Actions in Areas
- Routing Requests for Disk Files
- Configuring the Application Server
- Defining Routes for Disk Files
- Bypassing the Routing System
- URL Schema Best Practices
- Make Your URLs Clean and Human-Friendly
- GET and POST : Pick the Right One

Controllers and Actions

- Introducing the Controller
- Creating a Controller with IController
- Creating a Controller by Deriving from the Controller Class
- Receiving Request Data
- Getting Data from Context Objects

- Using Action Method Parameters
- Producing Output
- Understanding Action Results
- Returning HTML by Rendering a View
- Passing Data from an Action Method to a View
- Performing Redirections
- Returning Errors and HTTP Codes

Day4:

Filters

- Using Filters
- Introducing the Filter Types
- Applying Filters to Controllers and Action Methods
- Using Authorization Filters
- Applying the Custom Authorization Filter
- Using the Built-in Authorization Filter
- Using Authentication Filters
- Understanding the IAuthenticationFilter Interface
- Implementing the Authentication Check
- Combining Authentication and Authorization Filters
- Handling the Final Challenge Request
- Using Exception Filters
- Creating an Exception Filter
- Applying the Exception Filter
- Using a View to Respond to an Exception
- Using the Built-in Exception Filter
- Using Action Filters
- Implementing the OnActionExecuting Method
- Implementing the OnActionExecuted Method
- Using Result Filters
- Using the Built-in Action and Result Filter Class
- Using Other Filter Features
- Filtering Without Attributes
- Using Global Filters
 - Ordering Filter Execution
 - Overriding Filters

Controller Extensibility

- Creating a Custom Controller Factory
- Dealing with the Fallback Controller
- Instantiating Controller Classes
- Implementing the Other Interface Methods
- Registering a Custom Controller Factory
- Working with the Built-in Controller Factory
- Prioritizing Namespaces
- Customizing `DefaultControllerFactory` Controller Instantiation
- Creating a Custom Action Invoker
- Using the Built-in Action Invoker
- Using a Custom Action Name
- Using Action Method Selection
- Improving Performance with Specialized Controllers
- Using Sessionless Controllers
- Using Asynchronous Controllers

Views

- Creating a Custom View Engine
- Creating a Custom `IView`
- Creating an `IViewEngine` Implementation
- Registering a Custom View Engine
- Testing the View Engine
- Working with the Razor Engine
- Preparing the Example Project
- Understanding Razor View Rendering
- Configuring the View Search Locations
- Adding Dynamic Content to a Razor View
- Using Layout Sections
- Using Partial Views
- Using Child Actions

Helper Methods

- Creating Custom Helper Methods

- Creating an Inline Helper Method
- Creating an External Helper Method
- Managing String Encoding in a Helper Method
- Using the Built-In Form Helper Methods
- Creating Form Elements
- Specifying the Route Used by a Form
- Using Input Helpers
- Creating Select Elements

Templated Helper Methods

- Using Templated Helper Methods
- Generating Label and Display Elements
- Using Whole-Model Templated Helpers
- Using Model Metadata
- Using Metadata to Control Editing and Visibility
- Using Metadata for Labels
- Using Metadata for Data Values
- Using Metadata to Select a Display Template
- Applying Metadata to a Buddy Class
- Working with Complex Type Properties
- Customizing the Templated View Helper System
- Creating a Custom Editor Template
- Creating a Generic Template
- Replacing the Built-in Templates

Day5:

URL and Ajax Helper Methods

- Creating Basic Links and URLs
- Using MVC Unobtrusive Ajax
- Creating the Synchronous Form View
- Preparing the Project for Unobtrusive Ajax
- Creating an Unobtrusive Ajax Form
- Preparing the Controller
- Creating the Ajax Form
- Understanding How Unobtrusive Ajax Works

- Setting Ajax Options
- Ensuring Graceful Degradation
- Providing the User with Feedback While Making an Ajax Request
- Prompting the User Before Making a Request
- Creating Ajax Links
- Ensuring Graceful Degradation for Links
- Working with Ajax Callbacks
- Working with JSON
- Adding JSON Support to the Controller
- Processing JSON in the Browser
- Preparing Data for Encoding
- Detecting Ajax Requests in the Action Method

Model Binding

- Understanding Model Binding
- Using the Default Model Binder
- Binding to Simple Types
- Binding to Complex Types
- Binding to Arrays and Collections
- Manually Invoking Model Binding
- Dealing with Binding Errors
- Customizing the Model Binding System
- Creating a Custom Value Provider
- Creating a Custom Model Binder
- Registering the Custom Model Binder

Model Validation

- Explicitly Validating a Model
- Displaying Validation Errors to the User
- Displaying Validation Messages
- Displaying Property-Level Validation Messages
- Using Alternative Validation Techniques
- Performing Validation in the Model Binder
- Specifying Validation Rules Using Metadata
- Defining Self-Validating Models

- Performing Client-Side Validation
- Enabling Client-Side Validation
- Using Client-Side Validation
- Understanding How Client-Side Validation Works
- Performing Remote Validation

Bundles

- Profiling Script and Style Sheet Loading
- Using Script and Style Bundles
- Adding the NuGet Package
- Defining the Bundles
- Applying Bundles
- Optimizing the JavaScript and CSS Files

Web API and Single-page Applications

- Understanding Single-page Applications
 - Creating the Model
 - Adding the NuGet Packages
 - Adding the Controller
 - Adding the Layout and Views
 - Setting the Start Location and Testing the Example Application
 - Using Web API
 - Creating the Web API Controller
 - Testing the API Controller
 - Understanding How the API Controller Works
 - Understanding API Controller Action Selection
 - Mapping HTTP Methods to Action Methods
 - Using Knockout for Single-page Applications
 - Adding the JavaScript Libraries to the Layout
 - Implementing the Summary
 - Implementing the Create Feature
 - Completing the Application
 - Simplify the Home Controller
 - Manage Content Visibility
-

Duration and pricing

- *Full-time* over 5 days (R9995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
 - [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course price.
 - Prices exclude Vat for Vat-registered companies

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950 including the course material and guidance session.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

C# 6 and .NET Core 1.0

C# 6 and .NET Core 1.0

Prerequisites / Further Training

- [Introduction to Programming](#)
- [Beginner C#.NET](#) (C# Fundamentals and Object – Orientation)
- *Advanced C#.NET (Building practical Apps)*
- [SQL Querying](#) (Database design and querying)
- [ASP.NET CORE MVC](#) (Web App Development)
- ASP.NET Web API (More Web App Development)
- Entity Framework

Also have a look at our [.NET Bootcamp](#) and save

Intended Audience

This course provides a comprehensive foundation in the C# programming language and the core aspects of the .NET platform plus overviews of technologies built on top of C# and .NET (ADO.NET and Entity Framework, Windows Communication Foundation [WCF], Windows Presentation Foundation [WPF], and ASP.NET [WebForms, MVC, WebAPI]). Once you digest the information presented in course, you'll be in a perfect position to apply this knowledge to your specific programming assignments, and you'll be well equipped to explore the .NET universe further.

Day 1 topics are covered in review mode as these are covered in detail on the [C#.NET Beginner Course](#) – it is required and therefore assumed that you are up to speed on at least 66.7% of day 1 topics.

After this course you should be able to

- Understand the .NET Core platform and C# 6.
- Discover the ins and outs of the leading .NET technology.
- Find complete coverage of XAML, .NET Core and Visual Studio 2015 together with an overview of the new Windows Runtime
- Students will gain a solid foundation of object-oriented development techniques, attributes and reflection, generics and collections as well as advanced topics such as CIL opcodes and emitting dynamic assemblies.

Alignment

On the same level as Microsoft : [20483B](#) . View the [MCPD Course Schedules](#)

Further Training

[00 Analysis and Design](#)

[Design Patterns](#)

[ASP.NET Forms](#)

Course Material

Up to date course material is provided

Course info

DAY 1:

.NET CORE AND WINDOWS RUNTIME

- .NET Compiler Platform
- Testing
- Diagnostics and Application Insights
- Tasks and Parallel Programming
- Task Synchronization

DAY 2:

- Files and Streams
- Continues
- Security
- Networking
- Composition
- XML and JSON
- Localization

DAY 3 : WINDOWS APPS

- Core XAML
- Styling XAML Apps
- Patterns with XAML Apps
- Windows Apps: User Interfaces
- Advanced Windows Apps
- Windows Desktop Applications with WPF
- Creating Documents with WPF
- Deploying Windows Apps

DAY 4:

- WEP APPS AND SERVICES
- ADO.NET
- Entity Framework Core
- Windows Services
- ASP.NET Core

DAY 5:

- ASP.NET MVC
- ASP.NET Web API
- WebHooks and SignalR
- Windows Communication Foundation
- Deploying Websites and Services

Duration and pricing

- Full-time over 5 days (R9995)

- Part-time over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- Part-time over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For part-time courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
- “[Distance Learning](#)” over up to 3 months (R9995)
- International exams are not included in the course price.
- Prices exclude Vat for Vat-registered companies

Certificate

- Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
- You may sit for our competency assessment test and on passing you will obtain our competency certificate.
- Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

C#.NET Advanced

C#.NET 6 on .NET 4.6

Prerequisites / Further Training

- [Introduction to Programming](#)
- [Beginner C#.NET](#) (C# Fundamentals and Object – Orientation)
- *Advanced C#.NET (Building practical Apps)*

- [SQL Querying](#) (Database design and querying)
- [ASP.NET MVC](#) (Web App Development)
- ASP.NET Web API (More Web App Development)
- Entity Framework

Also have a look at our [.NET Bootcamp](#) and save

Intended Audience

This course provides a comprehensive foundation in the C# programming language and the core aspects of the .NET platform plus overviews of technologies built on top of C# and .NET (ADO.NET and Entity Framework, Windows Communication Foundation [WCF], Windows Presentation Foundation [WPF], and ASP.NET [WebForms, MVC, WebAPI]). Once you digest the information presented in course, you'll be in a perfect position to apply this knowledge to your specific programming assignments, and you'll be well equipped to explore the .NET universe further.

Day 1 topics are covered in review mode as these are covered in detail on the [C#.NET Beginner Course](#) – it is required and therefore assumed that you are up to speed on at least 66.7% of day 1 topics.

After this course you should be able to

- Understand the .NET 4.6 platform and C# 6.
- Discover the ins and outs of the leading .NET technology.
- Find complete coverage of XAML, .NET 4.6 and Visual Studio 2015 together with an overview of the new Windows Runtime
- Students will gain a solid foundation of object-oriented development techniques, attributes and reflection, generics and collections as well as advanced topics such as CIL opcodes and emitting dynamic assemblies.

Alignment

On the same level as Microsoft : [20483B](#) . View the [MCPD Course Schedules](#)

Further Training

[00 Analysis and Design](#)

[Design Patterns](#)

[ASP.NET Forms](#)

Course Material

Up to date course material is provided

Course info

Day 1

Day 1 topics are covered in review mode as these are covered in detail on the [C#.NET Beginner Course](#) – it is required and assumed that you are up to speed on at least 66.7% of day 1 topics.

Overview of C#.NET Beginner

- An Initial Look at the .NET Platform
- Introducing the Building Blocks of the .NET Platform (the CLR, CTS, and CLS)
- Additional .NET-Aware Programming Languages
- An Overview of .NET Assemblies
- Understanding the Common Type System
- Understanding the Common Language Specification
- Understanding the Common Language Runtime
- The Assembly/Namespace/Type Distinction
- Exploring an Assembly Using ildasm.exe
- The Platform-Independent Nature of .NET

- Building C# Applications on the Windows OS
- Building .NET Applications Beyond the Windows OS

- The Anatomy of a Simple C# Program
- An Interesting Aside: Some Additional Members of the

- System.Environment Class
- The System.Console Class
- System Data Types and Corresponding C# Keywords
- Working with String Data
- Narrowing and Widening Data Type Conversions
- Understanding Implicitly Typed Local Variables
- C# Iteration Constructs
- Decision Constructs and the Relational/Equality Operators

- Methods and Parameter Modifiers
- Understanding C# Arrays
- Understanding the enum Type
- Understanding the Structure (aka Value Type)
- Understanding Value Types and Reference Types
- Understanding C# Nullable Types

Encapsulation

- Introducing the C# Class Type
- Understanding Constructors
- The Role of the this Keyword
- Understanding the static Keyword
- Defining the Pillars of OOP
- C# Access Modifiers
- The First Pillar: C#'s Encapsulation Services
- Understanding Automatic Properties
- Understanding Object Initialization Syntax
- Working with Constant Field Data
- Understanding Partial Classes

Inheritance / Polymorphism

- The Basic Mechanics of Inheritance
- Revising Visual Studio Class Diagrams
- The Second Pillar of OOP: The Details of Inheritance
- Programming for Containment/Delegation
- The Third Pillar of OOP: C#'s Polymorphic Support
- Understanding Base Class/Derived Class Casting Rules

- The Master Parent Class: System.Object

Exceptions

- Ode to Errors, Bugs, and Exceptions
- The Role of .NET Exception Handling
- The Simplest Possible Example
- Configuring the State of an Exception
- System-Level Exceptions (System.SystemException)
- Application-Level Exceptions (System.ApplicationException)
- Processing Multiple Exceptions
- Debugging Unhandled Exceptions Using Visual Studio

Interfaces

- Understanding Interface Types
- Defining Custom Interfaces
- Implementing an Interface
- Invoking Interface Members at the Object Level
- Interfaces As Parameters
- Interfaces As Return Values
- Arrays of Interface Types
- Implementing Interfaces Using Visual Studio
- Explicit Interface Implementation
- Designing Interface Hierarchies
- The IEnumerable and IEnumerator Interfaces
- The ICloneable Interface
- The IComparable Interface

Day 2

The core C# Advanced Course starts here

Collections and Generic

- The Motivation for Collection Classes
- The Problems of Nongeneric Collections
- The Role of Generic Type Parameters
- The System.Collections.Generic Namespace

- The System.Collections.ObjectModel Namespace
- Creating Custom Generic Methods
- Creating Custom Generic Structures and Classes
- Constraining Type Parameters

Delegates / events / lambdas

- Understanding the .NET Delegate Type
- The Simplest Possible Delegate Example

Sending Object State Notifications Using Delegates

- Understanding Generic Delegates
- Understanding C# Events
- Understanding C# Anonymous Methods
- Understanding Lambda Expressions

Advanced language features

- Understanding Indexer Methods
- Understanding Operator Overloading
- Understanding Custom Type Conversions
- Understanding Extension Methods
- Understanding Anonymous Types
- Working with Pointer Types

LINQ to Objects

- LINQ-Specific Programming Constructs
- Understanding the Role of LINQ
- Applying LINQ Queries to Primitive Arrays
- Returning the Result of a LINQ Query
- Applying LINQ Queries to Collection Objects
- Investigating the C# LINQ Query Operators
- The Internal Representation of LINQ Query Statements

Day 3

Object Lifetime

- Classes, Objects, and References
- The Basics of Object Lifetime
- The Role of Application Roots
- Understanding Object Generations
- Concurrent Garbage Collection Prior to .NET 4.0
- Background Garbage Collection Under .NET 4.0 and Beyond
- The System.GC Type
- Building Finalizable Objects
- Building Disposable Objects
- Building Finalizable and Disposable Types
- Understanding Lazy Object Instantiation

Assembly configuration

- Defining Custom Namespaces
- The Role of .NET Assemblies
- Understanding the Format of a .NET Assembly
- Building and Consuming Custom Class Library
- Understanding Private Assemblies
- Understanding Shared Assemblies
- Consuming a Shared Assembly
- Configuring Shared Assemblies
- Understanding Publisher Policy Assemblies
- Understanding the <codeBase> Element
- The System.Configuration Namespace
- The Configuration File Schema Documentation

Reflection and attributes

- The Necessity of Type Metadata
- Understanding Reflection
- Building a Custom Metadata Viewer
- Dynamically Loading Assemblies
- Reflecting on Shared Assemblies
- Understanding Late Binding
- Understanding the Role of .NET Attributes
- Building Custom Attributes
- Assembly-Level Attributes

- Reflecting on Attributes Using Early Binding
- Reflecting on Attributes Using Late Binding
- Putting Reflection, Late Binding, and Custom Attributes in Perspective
- Building an Extendable Application

Dynamic types / DLR

- The Role of the C# dynamic Keyword
- The Role of the Dynamic Language Runtime
- Simplifying Late-Bound Calls Using Dynamic Types
- Simplifying COM Interoperability Using Dynamic Data
- COM Interop Using C# Dynamic Data

Processes, appdomains

- The Role of a Windows Process
- Interacting with Processes Under the .NET Platform
- Understanding .NET Application Domains
- Interacting with the Default Application Domain
- Creating New Application Domains
- Understanding Object Context Boundaries
- Summarizing Processes, AppDomains, and Context

CIL

- Motivations for Learning the Grammar of CIL
- Examining CIL Directives, Attributes, and Opcodes
- Pushing and Popping: The Stack-Based Nature of CIL
- Understanding Round-Trip Engineering
- Understanding CIL Directives and Attributes
- .NET Base Class Library, C#, and CIL Data Type Mappings
- Defining Type Members in CIL
- Examining CIL Opcodes
- Building a .NET Assembly with CIL
- Understanding Dynamic Assemblies

Day 4

Threading

- The Process/AppDomain/Context/Thread Relationship
- A Brief Review of the .NET Delegate
- The Asynchronous Nature of Delegates
- Invoking a Method Asynchronously
- The System.Threading Namespace
- The System.Threading.Thread Class
- Manually Creating Secondary Threads
- The Issue of Concurrency
- Programming with Timer Callbacks
- Understanding the CLR ThreadPool
- Parallel Programming Using the Task Parallel LibraryParallel LINQ Queries (PLINQ)
- Asynchronous Calls with the async Keyword

File IO / Object Serialization

- Exploring the System.IO Namespace
- The Directory(Info) and File(Info) Types
- Working with the DirectoryInfo Type
- Working with the Directory Type
- Working with the DriveInfo Class Type
- Working with the FileInfo Class
- Working with the File Type
- The Abstract Stream Class
- Working with StreamWriters and StreamReaders
- Working with StringWriters and StringReaders
- Working with BinaryWriters and BinaryReaders
- Watching Files Programmatically
- Understanding Object Serialization
- Configuring Objects for Serialization
- Choosing a Serialization Formatter
- Serializing Objects Using the BinaryFormatter
- Serializing Objects Using the SoapFormatter
- Serializing Objects Using the XmlSerializer
- Serializing Collections of Objects
- Customizing the Soap/Binary Serialization Process

ADO.NET Part 1

- A High-Level Definition of ADO.NET
- Understanding ADO.NET Data Providers
- Additional ADO.NET Namespaces
- The Types of the System.Data Namespace
- Abstracting Data Providers Using Interfaces
- Creating the AutoLot Database
- The ADO.NET Data Provider Factory Model
- Understanding the Connected Layer of ADO.NET
- Working with Data Readers
- Building a Reusable Data Access Library
- Creating a Console UI-Based Front End
- Understanding Database Transactions

ADO.NET Part 2

- Understanding the Disconnected Layer of ADO.NET
- Understanding the Role of the DataSet
- Working with DataColumnns
- Working with DataRowns
- Working with DataTablens
- Binding DataTable Objects to Windows Forms GUIs
- Working with Data Adapters
- Adding Disconnected Functionality to AutoLotDAL.dll
- Multitabled DataSet Objects and Data Relationships
- The Windows Forms Database Designer Tools
- Isolating Strongly Typed Database Code into a Class Library
- Programming with LINQ to DataSet

Entity Framework

- Understanding the Role of the Entity Framework
- Code First from an Existing Database
- Using the Model Classes in CodeHandling Database Changes
- AutoLotDAL Version 4
- Test-Driving AutoLotDAL

- Entity Framework Migrations
- Revisiting the Transaction Test
- Concurrency
- Interception
- ObjectMaterialized and SavingChanges Events
- Deploying to SQL Server

LINQ to XML

- A Tale of Two XML APIs
- Members of the System.Xml.Linq Namespace
- Working with XElement and XDocument
- Manipulating an In-Memory XML Document

Day 5

Introducing Windows Communication Foundation

- A Potpourri of Distributed Computing APIs
- The Role of WCF
- Investigating the Core WCF Assemblies
- The Visual Studio WCF Project Templates
- The Basic Composition of a WCF Application
- The ABCs of WCF
- Building a WCF Service
- Hosting the WCF Service
- Building the WCF Client Application
- Simplifying Configuration Settings
- Using the WCF Service Library Project Template
- Hosting the WCF Service Within a Windows Service
- Invoking a Service Asynchronously from the Client
- Designing WCF Data Contracts

Windows Presentation Foundation

- The Motivation Behind WPF
- The Various Flavors of WPF
- Investigating the WPF Assemblies
- Building a WPF Application Without XAML

- Building a WPF Application Using Only XAML
- Transforming Markup into a .NET Assembly
- Understanding the Syntax of WPF XAML
- Building a WPF Application Using Code-Behind Files
- Building WPF Applications Using Visual Studio
- Building a Custom XAML Editor with Visual Studio

Programming with WPF Controls

- A Survey of the Core WPF Controls
- A Brief Review of the Visual Studio WPF Designer
- Controlling Content Layout Using Panels
- Building a Window's Frame Using Nested Panels
- Understanding WPF Commands
- Understanding Routed Events
- A Deeper Look at WPF APIs and Controls
- Building the Ink API Tab
- Introducing the Documents API
- Building the Documents Tab
- Introducing the WPF Data-Binding Model
- Understanding the Role of Dependency Properties
- Building a Custom Dependency Property

WPF Graphics Rendering Services

- Understanding WPF's Graphical Rendering Services
- Rendering Graphical Data Using Shapes
- WPF Brushes and Pens
- Applying Graphical Transformations
- Working with the Visual Studio Transform Editor
- Rendering Graphical Data Using Drawings and Geometries
- Working with Vector Images
- Rendering Graphical Data Using the Visual Layer

WPF Resources, Animations, Styles, and Templates

- Understanding the WPF Resource System
- Working with Object (Logical) Resources
- Understanding WPF's Animation Services

- Authoring Animations in XAML
- Understanding the Role of WPF Styles
- Logical Trees, Visual Trees, and Default Templates
- Building a Control Template with the Trigger Framework

Notifications, Commands, Validation, and MVVM

- Introducing Model-View-ViewModel
- The WPF Binding Notification System
- Validation
- Using Data Annotations
- Creating Custom Commands
- Fully Implementing MVVM
- Updating AutoLotDAL for MVVM
- Full MVVM Example

• Duration and pricing

- *Full-time* over 5 days (R9995)
 - *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
 - *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
 - [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course price.
 - Prices exclude Vat for Vat-registered companies

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

C#.NET Beginner

Prerequisites / Further Training

You should not be a complete beginner for this course. If you cannot [pass this test](#), you must do [Intro to Programming with Java Script](#) or [Intro to Programming with Python](#) first.

- *Beginner C#.NET (C# Fundamentals and Object – Orientation)*
- [Advanced C#.NET](#) (Building practical Apps)
- [SQL Querying](#) (Database design and querying)
- [ASP.NET MVC](#) (Web App Development)
- ASP.NET Web API (More Web App Development)
- Entity Framework

Also have a look at our [.NET Bootcamp](#) and save

Intended Audience

This course is for anyone at the Intro to Programming level who want to learn how to program in C# using the .NET framework

After this course you should be able to

Create a fun arcade game and build games and other projects

Further Training

Recommended – [C# Advanced](#)

Alignment

Together, the Beginner and Advanced C#.NET are on the same level as Microsoft : [10266A](#).

Course Material

Up to date course material is provided

Course info

Start Building With c#: Build something cool, fast!

- Why you should learn C#
- C# and the Visual Studio IDE
- Debugging

It's all Just Code

- Anatomy of a program
- Two classes can be in the same namespace
- Variables
- C# uses familiar math symbols
- Use the debugger to see your variables change
- Loops
- if/else statements make decisions
- Build an app from the ground up

Day 2

Objects: Get Oriented!

- Classes
- Methods

- Objects
- An example solution
- Properties
- Instances
- Naming
- Class diagrams
- There's an easier way to initialize objects
- A few ideas for designing intuitive classes

Types and References

- The variable's type determines what kind of data it can store
- Casting
- C# does some casting automatically
- When you call a method, the arguments must be compatible with the types of the parameters
- References are like labels for your object
- If there aren't any more references, your object gets garbage-collected
- Multiple references and their side effects
- Arrays can contain a bunch of reference variables, too
- Objects use references to talk to each other
- Controls are objects, just like any other object

Day 3

Encapsulation

- Building an Event – planning app
- Each option should be calculated individually
- It's easy to accidentally misuse your objects
- Encapsulation means keeping some of the data in a class private
- Use encapsulation to control access to your class's methods and fields
- Private fields and methods can only be accessed from inside the class

- A few ideas for encapsulating classes
- Encapsulation keeps your data pristine
- Properties make encapsulation easier
- Automatic properties
- Use a constructor to initialize private fields

Inheritance

- Another event-planning type of class
- When your classes use inheritance, you only need to write your code once
- Build up your class model by starting general and getting more specific
- Let's design a zoo simulator
- Use inheritance to avoid duplicate code in subclasses
- Create the class hierarchy
- Every subclass extends its base class
- Use a colon to inherit from a base class
- We know that inheritance adds the base class fields, properties, and methods to the subclass...
- Method overriding
- Use the `override` and `virtual` keywords to inherit behavior
- A subclass can access its base class using the `base` keyword
- When a base class has a constructor, your subclass needs one, too

Day 4

Interfaces and Abstract Classes

- An interface tells a class that it must implement certain methods and properties
- Use the `interface` keyword to define an interface
- Upcasting works with both objects and interfaces
- Downcasting lets you turn your appliance back into a coffee maker

- Upcasting and downcasting work with interfaces, too
- There's more than just public and private
- Access modifiers change visibility
- Some classes should never be instantiated
- An abstract class is like a cross between a class and an interface
- An abstract method doesn't have a body
- Polymorphism means that one object can take many different forms

Enums and Collections

- Enums let you work with a set of valid values
- Enums let you represent numbers with names
- Lists make it easy to store collections of...anything
- Lists are more flexible than arrays
- Lists shrink and grow dynamically
- Generics can store any type
- Collection initializers are similar to object initializers
- Lists are easy, but SORTING can be tricky
- IComparable helps your list sort
- Use IComparer tells your List how to sort
- IComparer can do complex comparisons
- Overriding a ToString() method lets an object describe itself
- You can upcast an entire list using IEnumerable
- You can build your own overloaded methods
- Use a dictionary to store keys and values
- A queue is FIFO—First In, First Out
- A stack is LIFO—Last In, First Out

Day 5

Reading and Writing Files

- Use Stream.Read() to read bytes from a stream
- StreamWriter

- StreamReader
- You can read and write serialized files manually, too
- Use a BinaryWriter to write binary data
- C# can use byte arrays to move data around
- .NET uses Unicode to store characters and text
- Serialization
- Use a switch statement to choose the right option
- Writing files usually involves making a lot of decisions
- Avoid filesystem errors with using statements
- IDisposable makes sure your objects are disposed of properly
- Use file dialogs to open and save files (all with just a few lines of code)
- Use the built-in File and Directory classes to work with files and directories
- Dialog boxes are objects, too
- Dialog boxes are just another WinForms control
- Reading and writing using two objects
- .NET streams to read and write data

Duration and pricing

- *Full-time* over 5 days (R9995)
- *Part-time* over 4 weeks (2 nights per week, 3 hour sessions) (R11995)
- *Part-time* over 8 Saturdays, 3 hour sessions (R11995)
- Please note : For *part-time* courses we do not have a fixed schedule and you will be placed on a waiting list until we get a group of 4+ together. Please book with no dates on the bookings form. This will automatically put you on the waiting list. We will confirm with you as soon as we have a part-time group together.
- [Distance-learning](#) over up to 3 months (R9995)
- International exams are not included in the course price.
- Prices exclude Vat for Vat-registered companies
- **Monthly payment options for Distance Learning / Self**

Study

If you want to pay the course on a monthly basis, we divide the course in 4 sections – one per month. You then have to complete a quarter of the course per month. The payments are as follows:

- R3333 registration fee and covering the first month and section 1
- R3333- month 2 and section 2
- R3333 – month 3 the last section

▪ **Certificate**

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950, including the course material and a guidance session.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try

using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

C++ Advanced (with QT)

This Advanced C++ with QT Training course will teach you how to create hybrid desktop/Internet applications, threading, rich text handling, Qt's graphics/view architecture, and Qt's model/view architecture and many other topics are covered. Besides each chapter's main subject, lots of other classes, methods, and techniques are used wherever they make sense, to show as many Qt features as possible.

Prerequisites

You should be solidly at the [C++ Beginner](#) level before doing this course.

Further Training

[HTML5 / CSS3 Responsive Web Design](#) (if you want to learn more about creating responsive front-ends).

How about JavaScript, Angular, Ionic ?

Course Material

Included

Course Contents

Hybrid Desktop/Internet Applications

- Internet-Aware Widgets
- Using WebKit

- A Generic Web Browser Window Component
- Creating Web Site-Specific Applications
- Embedding Qt Widgets in Web Pages

Audio and Video

- Using QSound and QMovie
- The Phonon Multimedia Framework
- Playing Music
- Playing Videos

Model/View Table Models

- Qt's Model/View Architecture
- Using QStandardItemModels for Tables
- Changing a Table Model through the User Interface
- A QStandardItemModel Subclass for Tables
- A QSortFilterProxyModel to Filter Out Duplicate Rows
- A QSortFilterProxyModel to Filter In Wanted Rows
- Creating Custom Table Models
- Changing a Table Model through the User Interface
- A Custom QAbstractTableModel Subclass for Tables
- The QAbstractItemModel API Methods for Tables
- Methods to Support Saving and Loading Table Items

Model/View Tree Models

- Using QStandardItemModels for Trees
- Changing a Tree Model through the User Interface
- A QStandardItem Subclass for Tree Items
- A QStandardItemModel Subclass for Trees
- Creating Custom Tree Models
- Changing a Tree Model through the User Interface
- A Custom Item Class for Tree Items
- A Custom QAbstractItemModel Subclass for Trees
- The QAbstractItemModel API for Trees
- The QAbstractItemModel API for Drag and Drop
- Methods for Saving and Loading Tree Items

Model/View Delegates

- Datatype-Specific Editors
- Datatype-Specific Delegates
- A Read-Only Column or Row Delegate
- An Editable Column or Row Delegate
- Model-Specific Delegates

Model/View Views

- QAbstractItemView Subclasses
- Model-Specific Visualizing Views
- The Visualizer Widget
- The Visualizer's Aggregated Header Widget
- The Visualizer's Aggregated View Widget

Threading with QtConcurrent

- Executing Functions in Threads
- Using QtConcurrent::run()
- Using QRunnable
- Filtering and Mapping in Threads
- Using QtConcurrent to Filter
- Using QtConcurrent to Filter and Reduce
- Using QtConcurrent to Map

Threading with QThread

- Processing Independent Items
- Processing Shared Items

Creating Rich Text Editors

- Introducing QTextDocument
- Creating Custom Text Editors
- Completion for Line Edits and Comboboxes
- Completion and Syntax Highlighting for Text Editors
- Completion for Multi-line Editors
- Syntax Highlighting
- A Rich Text Single Line Editor

- Multi-line Rich Text Editing

Creating Rich Text Documents

- Exported QTextDocument File Quality
- Creating QTextDocuments
- Creating QTextDocuments with HTML
- Creating QTextDocuments with QTextCursor
- Exporting and Printing Documents
- Exporting QTextDocuments
- Exporting in PDF and PostScript Format
- Exporting in Open Document Format
- Exporting in HTML Format
- Exporting in SVG Format
- Exporting in Pixmap Formats
- Printing and Previewing QTextDocuments
- Painting Pages
- Painting PDF or PostScript
- Painting SVG
- Painting Pixmap

Creating Graphics/View Windows

- The Graphics/View Architecture
- Graphics/View Widgets and Layouts
- Introducing Graphics Items

Creating Graphics/View Scenes

- Scenes, Items, and Actions
- Creating the Main Window
- Saving, Loading, Printing, and Exporting Scenes
- Saving Scenes
- Loading Scenes
- Printing and Exporting Scenes
- Manipulating Graphics Items
- Adding Items
- Copying, Cutting, and Pasting Items
- Manipulating Selected Items

- Showing and Hiding the Guideline Grid
- Keeping the User Interface Up to Date
- Enhancing QGraphicsView
- Creating a Dock Widget Toolbox
- Creating Custom Graphics Items
- Enhancing QGraphicsTextItem
- Graphics Item Transformations
- Enhancing an Existing Graphics Item
- Creating a Custom Graphics Item from Scratch

The Animation and State Machine Frameworks

- Introducing the Animation Framework
- Introducing the State Machine Framework
- Combining Animations and State Machines

PROJECT

Duration and pricing

In [Price Group A](#)

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You may sit for our competency assessment test and on passing you will obtain our competency certificate.
3. Our competency assessment can be booked and taken by someone who has not attended the course at a cost of R2950.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

C++ Beginner

C++ Beginner

Learn one of the most popular and widely-used languages in the world. Learn Object-Orientation and if you are interested in the international exam, our syllabus covers it.

Prerequisites

You should not be a complete beginner for this course. If you cannot [pass this test](#), you must do [Intro To Programming](#) first.

Alignment

C++ Certified Associate Programmer (CAP) from the C++ Institute

Further Training

[C++ Advanced](#)

Course Material

Included

Course Contents

DAY 1:

Setting Out to C++

- C++ Initiation
- C++ Statements
- More C++ Statements
- Functions
- Summary
- Chapter Review
- Programming Exercises

Dealing with Data

- Simple Variables
- The const Qualifier
- Floating-Point Numbers
- C++ Arithmetic Operators

Compound Types

- Introducing Arrays
- Strings
- Introducing the string Class
- Introducing Structures
- Unions
- Enumerations
- Pointers and the Free Store
- Pointers, Arrays, and Pointer Arithmetic
- Combinations of Types
- Array Alternatives

Loops and Relational Expressions

- Introducing for Loops
- The while Loop

- The do while Loop
- The Range-Based for Loop (C++11)
- Loops and Text Input
- Nested Loops and Two-Dimensional Arrays
- Summary
- Chapter Review
- Programming Exercises

DAY 2:

Branching Statements and Logical Operators

- The if Statement
- Logical Expressions
- The ctype Library of Character Functions
- The ?: Operator
- The switch Statement
- The break and continue Statements
- Number-Reading Loops
- Simple File Input/Output

Functions: C++'s Programming Modules

- Function Review
- Function Arguments and Passing by Value
- Functions and Arrays
- Functions and Two-Dimensional Arrays
- Functions and C-Style Strings
- Functions and Structures
- Functions and string Class Objects
- Functions and array Objects
- Recursion
- Pointers to Functions

Adventures in Functions

- C++ Inline Functions
- Reference Variables

- Default Arguments
- Function Overloading
- Function Templates

Memory Models and Namespaces

- Separate Compilation
- Storage Duration, Scope, and Linkage
- Namespaces

DAY 3:

Objects and Classes

- Procedural and Object-Oriented Programming
- Abstraction and Classes
- Class Constructors and Destructors
- Knowing Your Objects: The this Pointer
- An Array of Object
- Class Scope
- Abstract Data Types

Working with Classes

- Operator Overloading
- Time on Our Hands: Developing an Operator
- Overloading Example
- Introducing Friends
- Overloaded Operators: Member Versus Nonmember
- Functions
- More Overloading: A Vector Class
- Automatic Conversions and Type Casts for Classes

Classes and Dynamic Memory Allocation

- Dynamic Memory and Classes
- The New, Improved String Class
- Things to Remember When Using new
- in Constructors

- Observations About Returning Objects
- Using Pointers to Objects
- Reviewing Techniques
- A Queue Simulation

Class Inheritance

- Beginning with a Simple Base Class
- Inheritance: An Is-a Relationship
- Polymorphic Public Inheritance
- Static and Dynamic Binding
- Access Control: protected
- Abstract Base Classes
- Inheritance and Dynamic Memory Allocation
- Class Design Review

DAY 4:

Reusing Code in C++

- Classes with Object Members
- Private Inheritance
- Multiple Inheritance
- Class Templates

Friends, Exceptions, and More

- Friends
- Nested Classes
- Exceptions
- Runtime Type Identification
- Type Cast Operators

The string Class and the Standard

- Template Library
- The string Class
- Smart Pointer Template Classes
- The Standard Template Library

- Generic Programming
- Function Objects (a.k.a. Functors)
- Algorithms
- Other Libraries

Input, Output, and Files

- An Overview of C++ Input and Output
- Output with cout
- Input with cin
- File Input and Output
- Incore Formatting

DAY 5:

Visiting with the New C++ Standard

- C++11 Features Revisited
- Move Semantics and the Rvalue Reference
- New Class Features
- Lambda Functions
- Wrappers
- Variadic Templates
- More C++11 Features
- Language Change

PROJECT

—

Duration and pricing

In [Price Group A](#)

Certificate

Read about [our certificates here](#)

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Django

Intended Audience

This Django Training Course is for Python programmers who want to learn the best and most-widely used framework for developing Python-based Web applications.

Prerequisites / Further Training

Recommended sequence	Summary
Introduction to Programming.	Fundamentals of Computer Programming
Beginner Python	Python Language and OO Fundamentals
Advanced Python (Full-Stack)	Building Various Apps with Python

Django or Flask	<i>Python Web development in detail</i>
---------------------------------	---

Also have a look at our [Python Bootcamp](#)

After this course you should be able to

Quickly start a new Django project and establish a strong foundation for a growing site

- Define how your data is organized and create a SQLite database to manage it
- Quickly produce HTML with Django templates
- Create dynamic webpages with Django's URL patterns and views, including function views, class-based views, and generic views
- Enable efficient, reliable data input with Django Forms and custom form validations
- Understand the Model-View-Controller (MVC) architecture, compare it to Model-Template-Views, and gain a holistic understanding of Django's structure
- Write as little code as possible, simplify code reuse, and mitigate software decay by adhering to the Don't Repeat Yourself paradigm.
- Dive into Django source code to troubleshoot problems
- Extend site functionality with Django's contributed library
- Protect your site with user authentication and permissions
- Avoid security pitfalls such as SQL Injection, XSS, and CSRF
- Optimize site performance
- Develop complete Python-based Web applications from start to finish in Django.

Course Material

Supplied

Course Contents

Day 1

- Starting a New Django Project
- Building a Basic WebPage in Django
- Programming Django Models and Creating a SQLite Database
- Rapidly Producing HTML with Django Templates
- Webpages with Controllers, Views and URL Configurations
- Use Models, Templates, Views and URL Configurations to Create Links between Web Pages

Day 2

- User Input with Forms
- Displaying Forms in Templates
- Controlling forms in Views
- Migrations
- The Contact Us Webpage
- The Big Picture : Django's Core

Day 3

- Django's Contributed Libraries
- Pagination for Navigation
- Django Flatpages
- Static Content with Django
- Generic class-based views
- Advanced Generic class-based view usage

Day 4

- Basic Authentication
- Integrating Permissions
- Authentication extension
- Customised Authentication
- Admin Library
- Custom Managers and Querysets

Day 5

- Handling Behavior with Signals
- Performance Improvement of the website
- Building Custom Template Tags
- Adding RSS and Atom Feeds and a sitemap
- Deploy
- Starting a new Project Correctly

Duration and pricing

- Full-time over 5 days (R9995 excl VAT)
- Part-time over 4 weeks (2 nights per week, 3 hour sessions) (R10995 excl Vat)
- Part-time over 8 Saturdays, 3 hour sessions (R10995 excl Vat)
- [Distance-learning](#) over up to 3 months (R7995 excl Vat)

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You have the option to get the competency / academic certificate if you :
hand in a project (pre-approved) covering most of the topics in the book.

Schedule

On the calender on this page below.

If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our 'Contact Us' page.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by [clicking here](#).

Questions

Please [email us](#)

We are a member of the Python Software Foundation



Flask

Full-Stack Development in Python with Flask Training Course

We go through the framework's core functionality, and show you how to extend applications with advanced web techniques such as database migration and web service communication.

Rather than impose development guidelines as other frameworks do, Flask leaves the business of extensions up to you. If you have Python experience, this course shows you how to take advantage of that creative freedom.

Prerequisites / Further Training

Recommended sequence	Summary
Introduction to Programming.	Fundamentals of Computer Programming
Beginner Python	Python Language and 00 Fundamentals

Advanced Python (Full-Stack)	Building Various Apps with Python
Django or <i>Flask</i>	<i>Python Web development</i>

Also have a look at our [Python Bootcamp](#)

Intended Audience

Python programmers who want to learn the best and most-promising, new generation and upcoming framework for developing Python-based Web applications.

After this course you should be able to

- Know Flask's basic application structure and write an example app
- Work with must-have components—templates, databases, web forms, and email support
- Use packages and modules to structure a large application that scales
- Implement user authentication, roles, and profiles
- Build a blogging feature by reusing templates, paginating item lists, and working with rich text
- Use a Flask-based RESTful API to expose app functionality to smartphones, tablets, and other third-party clients
- Learn how to run unit tests and enhance application performance
- Explore options for deploying your web app to a production server

Course Material

Supplied

Course Contents

<p style="text-align: center;">Day 1</p> <ul style="list-style-type: none"> • Installation • Basic Application Structure <ul style="list-style-type: none"> • Templates • Web Forms <p style="text-align: center;">Day 2</p> <ul style="list-style-type: none"> • Databases <ul style="list-style-type: none"> • Email • Large Application Structure <p style="text-align: center;">Day 3</p> <ul style="list-style-type: none"> • User Authentication <ul style="list-style-type: none"> • User Roles • User Profiles 	<p style="text-align: center;">Day 4</p> <ul style="list-style-type: none"> • Blog Posts • Followers • User Comments <p style="text-align: center;">Day 5</p> <ul style="list-style-type: none"> • Application Programming Interfaces <ul style="list-style-type: none"> • Testing • Performance • Deployment
--	---

Duration and pricing

- Full-time over 5 days (R9995 excl VAT)
- Part-time over 4 weeks (2 nights per week, 3 hour sessions) (R10995 excl Vat)
- Part-time over 8 Saturdays, 3 hour sessions (R10995 excl Vat)
- [Distance-learning](#) over up to 3 months (R7995 excl Vat)

Certificate

1. Upon completion of this course we will issue you with attendance certificate to certify your attendance and / or completion of the prescribed minimum examples.
2. You have the option to get the competency / academic certificate if you :
hand in a project (pre-approved) covering most of the topics in the book.

Schedule

On the calender on this page below.

If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our 'Contact Us' page.

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by [clicking here](#).

Questions

Please [email us](#)

We are a member of the Python Software Foundation



Java Advanced

Prerequisites / Further Training

- [Java Beginner](#)
- [Beginning SQL](#)

Also have a look at our [Java Bootcamp](#)

Alignment

Oracle OCP Certification aligned to Oracle OCP Java Exam

After this course you should be able to

- Have a good understanding of programming and the building blocks of an OO programming language, with an

emphasis on JAVA.

- Build small apps in Java, making use of I/O, Networking, GUI
- Prepare for Oracle OCA and OCP exams

Course Material

Course Material Provided

Course Contents

Day 1:

Static Classes

- Static Methods,
- Static Variables
- Math Class
- Math Methods
- Static Imports

Wrapper classes

- Autoboxing
- Number formatting

Day 2

Exception Handling

- What if you need to call risky code?
- Checked Exceptions
- Flow control in try/catch blocks
- Exceptions are polymorphic
- When you don't want to handle an exception...
- Ducking (by declaring) only delays the inevitable

Nested Classes

- Getting GUI: A Very Graphic Story
- Getting a user event

- Listeners, Sources, and Events
- An inner class instance must be tied to an outer class instance
- How to make an instance of an inner class

Day 3:

I/O Fundamentals

- Saving Object state
- Implement Serializable
- Transient variables

Highlights of the Calendar API

- Getting an object that extends Calendar
- Working with Calendar objects

Date / Time API

- Moving backward and forward in time

Day 4:

Networking

- Connecting, Sending, and Receiving
- Network Sockets
- Writing a simple server
- Writing a Chat Client

Threads & Concurrency

- Thread class
- Runnable interface
- The Thread Scheduler
- Putting a thread to sleep
- Atomic methods
- Using an object's lock
- Synchronization

Day 5:

Collections and Generics

- Collections
- Sorting an ArrayList with Collections.sort()
- Generics and type-safety
- Comparable interface
- A Custom Comparator
- Collection API : lists, sets, maps
- HashSet
- HashMap
- Polymorphism wildcards

Packaging

- Command-line Java
- Classpath
- Jars

Duration and pricing

[Price Group A](#)

Certificate

[Read About Our Certificates](#)

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Java Beginner

This Beginner Java Training Course will give you the fundamentals of the Java Programming Language with an emphasis on OO. This course is aligned with the Oracle Certified Associate in Java Programming Certification.

Prerequisites / Further Training

You should not be a complete beginner for this course. If you cannot [pass this test](#), you must do [Intro To Programming](#) Course first.

Alignment

OCA: Oracle Certified Associate Java SE 8 Programmer Exam 1Z0-808

Intended Audience

- Intended for people who have some knowledge of programming and want to learn Java and OO
- **NB:** This is NOT an [Introduction to Programming](#) course.

After this course you should be able to

- Have a good understanding of programming and the building blocks of an OO programming language, with an emphasis on JAVA.
- Prepare for Oracle OCA Exam 1Z0-803

- Proceed to the [Advanced Java](#) Training Course

Course Material

Course Material Provided

Course Contents

Day 1

Introducing Java Technology

- Breaking the Surface
- The way Java works
- Code Structure in Java
- Anatomy of a class
- The main() method
- Netbeans IDE and Debugging

Loop and decision constructs

- Looping
- Conditional branching
- A trip to Objectville
- Inheritance,Overriding
- Class variables and methods
- Making your first object,Using main

Day 2

Primitives

- Know your Variables
- Declaring a variable
- Primitive types
- Java keywords

Objects

- Reference variables
- Object declaration and assignment
- Objects on the garbage collectible heap
- Arrays

- How Objects Behave
- Methods use object state
- Method arguments and return types
- Pass-by-value

Encapsulation

- Getters and Setters
- Encapsulation
- Using references in an array

Day 3

Arrays and ArrayLists

- Extra Strength Methods
- Building a one-dim ArrayList game
- Preparing to code
- Coding
- Random numbers<
- Using user-input
- For loops
- Casting primitives
- String conversion
- Using the Java Library
- Two Dimensional ArrayList Structures
- Enhancing the game
- Coding the game
- Boolean expressions
- Using the Java library (API)
- Using packages
- Using the HTML API docs and

Day 4

Polymorphism. Method Overloading

- Better Living in Objectville
- Understanding inheritance
- Designing an inheritance tree

- Avoiding duplicate code
- Overriding methods
- IS-A and HAS-A · What do you inherit from your superclass?
- What does inheritance really buy you?
- Polymorphism
- Rules for overriding
- Method overloading

Advanced OO Concepts

- Serious Polymorphism
- Some classes should not be instantiated
- Abstract classes
- Abstract methods
- Polymorphism in action
- Class Object
- Taking objects out of an arraylist
- Compiler checks the reference type
- Get in touch with your inner object
- Polymorphic references
- Casting an object reference (moving lower in the inheritance tree)
- Deadly Diamond of Death
- Using interfaces (the best solution)

Day 5

Garbage Collection

- Life and Death of an Object
- The stack and the heap
- Methods on the stack
- Where local variables live
- Where instance variables live
- The miracle of object creation

Constructors

- Constructors, Initializing state of a new Object
- Overloaded constructors
- Superclass constructors
- Invoking overloaded constructors using this()
- Life of an object, Garbage collection

Handling Errors

- Handling Errors
- Handling Exceptions

Duration and pricing

In [Price Group A](#)

Certificate.

Please read about our [certificates](#).

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#)

Schedule

On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.

Java EE

Prerequisites / Further Training

You should be past the [Java Advanced](#) level

Intended Audience

This course is intended for Java Programmers who have mastered Java JSE and wants a thorough introduction to JEE .

After this Course

You should be familiar with the latest facilities and features of Java EE . Create Web Applications with Java using best practices. Servlets, JSF, primeFaces, JPA, Session Beans, CDI, JMS, Web Services with JAX-WS, JAX-RS. Build a Java Web Application that utilise all these features.

Course Material

- Included in the course price.
- Based on Java EE 7

Course Contents

Day 1

Introduction to JavaServer Faces

- Developing our first JSF application
- Facelets templating
- Resource library contracts
- Composite components
- Faces flows

- HTML5 support

JSF Component Libraries

- Using PrimeFaces components in our JSF applications
- Using ICEfaces components in our JSF applications
- Using RichFaces components in our JSF applications

Day 2

Interacting with Databases through the Java Persistence API

- Creating our first JPA entity
- Automated generation of JPA entities
- Generating JSF applications from JPA entities

Implementing the Business Tier with Session Beans

- Introducing session beans
- Creating a session bean
- Accessing the bean from a client
- Session bean transaction management
- Implementing aspect-oriented programming with interceptors
- The EJB Timer service
- Generating session beans from JPA entities

Day 3

Contexts and Dependency Injection

- Introduction to CDI
- Qualifiers
- Stereotypes
- Interceptor binding types
- Custom scopes

Messaging with JMS and Message-driven Beans

- Introduction to JMS
- Creating JMS resources from NetBeans
- Implementing a JMS message producer
- Consuming JMS messages with message-driven beans

Day 4

Java API for JSON Processing

- The JSON-P object model API
- The JSON-P streaming API

Java API for WebSocket

- Examining the WebSocket code using samples included with NetBeans
- Building our own WebSocket applications

Day 5

RESTful Web Services with JAX-RS

- Generating a RESTful web service from an existing database
- Testing our RESTful web service
- Generating RESTful Java client code
- Generating RESTful JavaScript clients for our RESTful web services

SOAP Web Services with JAX-WS

- Introduction to web services
- Creating a simple web service
- Exposing EJBs as web services

Duration and pricing

[Price Group A](#)

Certificate

[Read About Our Certificates](#)

Bookings

You can download the course registration form on our home page or by clicking [here](#)

Brochure

You may download a pdf copy of this page by clicking on the pdf icon at the top of the page.

Questions

Please [email us](#) Schedule On the calendar below. If your browser doesn't display the calendar below, please click on [this link](#) or try using [Google Chrome](#), alternatively please enquire via our [Contact Us](#) page.