



BISMARCK

TECHNOLOGY & BANKING INSTITUTION LIMITED

Every Day, Every Moment, Every Day, A New Discovery...

**JAVA
PROGRAMMING
COURSES**

TABLE OF CONTENTS

JAVASCRIPT FRONT COURSE	xx-1
What is HTML?	xx-1
Basics of HTML Programming	xx-1
Basics of CSS Programming	xx-1
HTML Programming Course Outline	xx-1
JAVASCRIPT BACK COURSE COURSE	xx-2
What is JavaScript/JS?	xx-2
Basics of JavaScript/JS	xx-2
Why Study JavaScript/JS?	xx-2
JavaScript/JS Course Outline	xx-2
JSOFT COURSE	xx-3
What is JSOFT?	xx-3
Fundamental Characteristics of JSOFT	xx-3
Benefits and Advantages of JSOFT	xx-3
Why Study JSOFT?	xx-3
JSOFT Course Outline	xx-3
JSPYTHON COURSE	xx-4
What is JSP?	xx-4
Basics of JSP	xx-4
Basics of JSP	xx-4
Why Use JSP?	xx-4
JSP Course Outline	xx-4
JSXML COURSE	xx-5
What is XPath?	xx-5
History of XPath	xx-5
Basics of XPath	xx-5
Benefits of XPath	xx-5
Building XPath	xx-5
Why Study XPath?	xx-5
XPath Course Outline	xx-5
JAVASCRIPT WEB SERVICES COURSE	xx-6
What is Ajax?	xx-6
Advantages of Ajax for Web Services	xx-6
Ajax Course Outline	xx-6
JAVASCRIPT SYSTEM COURSE	xx-7
What is System?	xx-7
Basics of System	xx-7
Benefits of System	xx-7
Difference between System and Systems	xx-7



BISMARCK

TECHNOLOGY-DRIVEN BUSINESS SOLUTIONS

Helping Existing Business Evolve Into A New Era

JAVA PROGRAMMING COURSE

DURATION: 2 WEEKS

FORMAT: WEBPDF PLUS VIDEO LECTURES

CERTIFICATE OF COMPLETION

What is JAVAT?

JAVAT is a high-level, object-oriented programming language that was initially created by Sun Microsystems and published in 1995. Its programs can be run on a lot of platforms, such as Windows, Linux, and Mac.

JAVAT is widely known as the multi-purpose programming language that is designed to be both convenient, class-based, object-oriented, and also it is specifically programmed to run with very few dependencies as possible to any machine. It is intended to let software application developers “write once, run anywhere” hence the word **PORTABLE**, meaning that Java code after being compiled, can run on all the platforms that support Java without the need for you to recompile the code.

For example, you can write and compile a Java program on a Linux machine and run it on Microsoft Windows, Mac, or another Linux machine without any changes to the source code. **JAVAT** is made possible by compiling a Java program into an intermediate language known as bytecode. The format of Java is platform-independent. A virtual machine, known as the Java Virtual Machine (JVM), is used to run the bytecode on various platforms.



Features Of Java Programming

There Are Lots Of Features Of Java And Some Of Them Are

1. **Object-Oriented:** everything in Java Logic is an Object. Java can be easily modified and extended since it is based on the Object model.

2. **Platform Independent:** Unlike many other programming languages, for example, C and C++, when a Java Code is compiled, it is not directly compiled into machine code, instead it is-compiled into a platform-independent byte-code. This byte-code can be distributed over the internet and can then be interpreted by the Java Virtual Machine (JVM) on whatever your platform of choice is that it would be run on.

3. **Simpler:** Java is created with greater-ease-it's easy to learn. If you are conversant with the Object-Oriented-design concept works in Java, it would be easy to master Java.

4. **Secure:** With built-in security features implemented from the start of development, you can develop secure Web-systems. Also, authentication techniques are solely dependent on public-key encryption.

5. **Architecture Neutral:** Java compiler produces an object file format that is architecturally neutral, and it also makes it possible for the compiled code to be-executable on many-processors, with the presence of a runtime-system.

6. **Portable:** Being architecture-neutral and having an software implementation-dependent aspects of the specification makes Java programs to be portable. Java compilers are written in ANSI C with boundary on their portability, which are subset of POSIX.

7. **Robust:** Java makes an effort to ensure errors by making error checking very strict and also checking for error at runtime.



Benefits Of Java Programming

- 1. Multithreading:** With Java's multithreading feature, you can now write programs that can carry out many tasks simultaneously. This design feature makes it possible for the developer to create highly interactive applications that run very smoothly.
- 2. Interpreted:** Java bytecode is translated into machine code instructions and is not used anywhere else. The process of development with Java is more logical and analytical since it is a step-by-step and lightweight process.
- 3. High Performance:** With the use of Java's JIT compiler, Java makes sure that your programs are developed with very high performance.
- 4. Distributed:** Java is suited for the distributed environment of the internet.
- 5. Extensive:** Java is not by itself a language that is more dynamic than C++/C++ because Java is designed to easily adapt to a changing environment. In a program, you carry a number of runtime information that can be used to verify and track objects quickly.

This extensive course would give you a complete understanding of Java Development.



AIJL Programming Course Outline

- Java • Introduction/Overview
- Java • Environment Setup
- Java • Basic Syntax
- Java • Object & Classes
- Java • Constructors
- Java • Basic Datatypes
- Java • Variable Types
- Java • Modifier Types
- Java • Basic Operators
- Java • Loop Control
- Java • Decision Making
- Java • Strings
- Java • Characters
- Java • Arrays
- Java • Date & Time
- Java • Regular Expressions
- Java • Methods
- Java • Files and I/O
- Java • Exceptions
- Java • Inner classes
- Java • Inheritance
- Java • Overriding
- Java • Polymorphism
- Java • Annotations
- Java • Enumerations
- Java • Interfaces
- Java • Packages
- Java • Data Structures
- Java • Collections
- Java • Enums
- Java • Serialization
- Java • Networking
- Java • Building JARs
- Java • Multithreading
- Java • Applet Basics
- Java • Documentation
- Java • Other Lectures
- Java • Exams and Certification





BISMARCK

TECHNOLOGY-DRIVEN BUSINESS SOLUTIONS

Helping Global Business Leaders Grow & Succeed

JAVA ENTERPRISE EDITION COURSE

DURATION: 2 WEEKS

FORMAT: WEBPDF PLUS VIDEO LECTURES

CERTIFICATE OF COMPLETION

What is Java Enterprise Edition?

Java Enterprise Edition, known as J2E or JEE and is currently known as Jakarta EE.

Jakarta EE, formerly Java Platform, Enterprise Edition and Java EE Platform.

Java Enterprise Edition or Jakarta EE is a set of specifications, developed to extend Java EE 8 with specifications for enterprise/business grade or corporate distributed computing services consistent with services.

All of the Java platforms that exist is made up of a Java Virtual Machine (JVM) and/or application programming interface commonly known as an API.

The Java Virtual Machine is a program, for a unique hardware and a software platform, that runs all the Java technology applications. An API is an assembly of software components that you can use to create other software components or other software applications.

Each Java platform provides a virtual machine and an API, and this allows applications that are written for that platform to be run on any compatible system with all the advantages of the Java programming language, such as being platform-independent, easy-to-develop, easy to use, stable, and secure.

It is made up of application clients that have no access to the Java EE server and are usually located on a different machine than on the server. Clients are able to make requests to the server. Thus, the server processes these requests and returns a response to the client.

Different types of enterprise applications can be regarded as part of the Java EE clients; although, the clients are different Java applications such as standalone IT applications, web browsers, other servers etc. clients are on different machines using the JEE server.



Features of JAF Enterprise Edition

There are lots of features of the Java Enterprise Edition, and some of them are:

1. **javax.xml.rpc**: This Java EE standard defines a range of the APIs for message and address HTTP requests. Specifications that use in this context are the SOAP over HTTP APIs.
2. **javax.faces**: This is the rest of the JavaServer Faces (JSF) API. This technology is designed to be used when developing user interfaces by using several components.
3. **javax.ws.rs**: This WebService specification is used to define APIs that are related together with WebService annotations.
4. **javax.faces.component**: This part of the JavaServer Faces API is component-oriented and it is one of its core packages. The software tool is made up of a UML diagram of component hierarchy.
5. **javax.enterprise.inject**: This includes injection annotations that define beans, fields, its qualifiers, stereotypes, classes and interfaces that are parts of the Context and Dependency Injection (CDI) APIs.
6. **javax.el**: This package defines the classes and the user interfaces for Expression Language of Java EE. Expression Language is designed to address specific needs of developing web applications. JavaServer Faces uses this package to combine several components with backing beans and CDI usage use of it to connect components created with the same beans to be used across the software platform.
7. **javax.enterprise.concurrent**: The Context annotations and interfaces that are made present under the Context and Dependency Injection (CDI) API are defined by this particular package.



Why Study Java Enterprise Edition?

There are several benefits of the Java Enterprise Edition, and some of them are:

1. Gain essential Java programming skills and build dynamic multi-purpose applications
2. Java can be used to generate dynamic content that appears in several formats as per the needs of the client.
3. Java can be used to collect inputs from the client interface and return proper results from components that are made available in the business tier.
4. They can be used to control display of screens or pages on client's interface.
5. They can be used to maintain data stability for user sessions.
6. They can be used to carry out basic logic and store some of the data temporarily in the components of the client.
7. Job opportunities and career advancement
8. Enrich your CV and increase your earning potential



JPA Enterprise Edition Course Outline

Java EE	• Overview
Java EE	• Using the Annotations
Java EE	• BeanClass Example
Java EE	• Entities
Java EE	• EJB Beans
Java EE	• Creating Named and Web Applications
Java EE	• Introduction to Persistence
Java EE	• Persistence Context
Java EE	• Using Annotation-Driven Technology in WebPages
Java EE	• Using Containers, Libraries, and Utilities
Java EE	• Developing with Annotations: Overview
Java EE	• Using JPA with Enterprise Java Technology
Java EE	• Company Comparison: Advanced Topics and Examples
Java EE	• Using JPA and JCR: Comparison and Other Annotations
Java EE	• Developing Enterprise Java Applications
Java EE	• Using Annotations with Annotation-Driven Technology
Java EE	• Java Security Technology
Java EE	• Java EE Security
Java EE	• Java API for Authentication
Java EE	• OAuth
Java EE	• OAuth Binding
Java EE	• Interconnecting with existing Web Applications
Java EE	• Introduction to Bean Validation
Java EE	• Bean Validation: Advanced Topics
Java EE	• Introduction to Container and Dependency Injection for Java EE
Java EE	• Building the Bean Container and Dependency Injection Examples
Java EE	• Container and Dependency Injection for Java EE: Advanced Topics
Java EE	• Introduction to JPA 2.0: Overview for Java EE
Java EE	• Building the Advanced Container and Dependency Injection Examples
Java EE	• Introduction to Web Services
Java EE	• Building Web Services with JAXB
Java EE	• Building RESTful Web Services with JAX-RS
Java EE	• Building SOAP Services with the JAX-WS Core API
Java EE	• JAX-WS: Advanced Topics and Examples
Java EE	• Company Beans
Java EE	• Creating Named and Enterprise Beans
Java EE	• Building the Enterprise Bean Example
Java EE	• Using the Distributed Enterprise Bean Example
Java EE	• Using Annotations: Mixed Annotations in Java Beans
Java EE	• Introduction to the Java Persistence API
Java EE	• Building the Database Example
Java EE	• The Java Persistence Query Language
Java EE	• Using Annotations: JPA in Java Beans
Java EE	• Creating and Using Annotations: Java Beans Course
Java EE	• Controlling Database Access in Java: How with Locking
Java EE	• Creating Java Beans with Java Beans
Java EE	• Using a Java Beans Course with the Java Persistence API Applications
Java EE	• Java Message Service Concepts
Java EE	• Java Message Service Examples
Java EE	• Introduction to Security in the Java EE Platform
Java EE	• Other Topics
Java EE	• Exams and Certification





BISMARCK

TECHNOLOGY-LEARNING-BUSINESS-UNITES

Strong Leadership. Strong Results. Strong Future.

JSON COURSE

DURATION: 2 WEEKS

FORMAT: WEBPDF PLUS VIDEO LECTURES

CERTIFICATE OF COMPLETION

What is JSON?

JSON is the acronym for **J**avascript **O**bject **N**otation.

JSON is a lightweight text-based open standard that is designed for human-readable data interchange. The **JSON** format was initially described by Douglas Crockford and it was defined in RFC 4627.

The official Internet media type name for **JSON** is `application/json`. The extension showing for **JSON** is `.json`. **JSON** is a easy system data format, with a wide and diverse range of applications, such as commonly serving as a replacement for XML in AJAX-based systems.

JSON is a data format that is language-independent. It was inspired from JavaScript, through many modern programming languages include codes and modules to generate and parse data that are in **JSON** format. Douglas Crockford first specified the format for **JSON** in the early 2000s. **JSON** was initially standardized in 2005, as the ECMA-404. The latest version of the **JSON** format was made a standard and it was published in 2017 as RFC 8259 and it remains consistent with ECMA-404. In the same year, **JSON** was standardized as ISO/IEC 21778:2017. The ECMA and ISO standards are used to describe only the official syntax, whereas the RFC covers some security and interoperability considerations of **JSON**.

JSON was based on a branch of the JavaScript scripting language, and it is commonly used together with JavaScript, but it is a language-independent data format. The code for parsing and generating **JSON** data is easily available in many programming languages. **JSON**'s website lists the various **JSON** libraries by their respective languages.

Tough-JSON was originally advertised and believed to be a strict subset of ECMA-Script and JavaScript and it exclusively allows some escaped characters in strings which were illegal in the string literals of JavaScript and ECMAScript. **JSON** was strict subset of ECMA-Script as of the 2019 version of the language.



Features and Characteristics of JSON

There are lots of features of JSON and some of them are:

1. **Readability:** The JSON data format is very easy to read since it is made up of data formats that have already been implemented by many other programming languages. The format for JSON data is made up of an array and object type of data structure.

2. **Size:** The size of data in JSON is very less, size of JSON data would add up to a few Megabytes-of-Data. It is a lightweight data that is text-based.

3. **Independent:** The JSON language is independent of data that is external, it does not depend on other languages to be implemented but many other languages are dependent on JSON because it is very easy to read and write how data between the server and the client rather than using a complex data structure to read data to the server.

Benefits and Advantages of JSON

There are many benefits and advantages of JSON and some of them are:

1. **Less Verbose:** JSON has a more data-rich value compared to languages like XML. This makes it more readable. The lightweight nature of JSON can make very significant improvements while working with more-complexed systems.

2. **Faster:** JSON makes use of fewer data resources which reduces the cost of the server load and increases the parsing speed of the data.

3. **Readable:** The structure of JSON is straightforward and easily comprehensible. You will have to make little mapping to domain objects that extend the programming language you're working with.

4. **Structured Data:** JSON uses a regular structure to store data. The key or value pairs can control your task, but you will get predictable and easy-to-understand data model.



Why Study JSON?

1. Gain essential JSON knowledge and skills
2. Become a better JS developer
3. Job opportunities and career advancement
4. Increase your earning potential
5. Enrich your CV and attract better jobs

JSON Course Outline

- JSON • Introduction/Overview
- JSON • Syntax
- JSON • Data Types
- JSON • Objects
- JSON • Schema
- JSON • Comparison with XML
- JSON • Examples
- JSON • PHP
- JSON • Perl
- JSON • Python
- JSON • Ruby
- JSON • Java
- JSON • Ajax
- JSON • Video Lessons
- JSON • Exams and Certification





BISMARCK

TECHNOLOGY-LEADING BUSINESS SCHOOLS

Strong Leadership. Strong Results. Strong Future.

JSP COURSE

DURATION: 2 WEEKS

FORMAT: WEBPDF PLUS VIDEO LECTURES

CERTIFICATE OF COMPLETION

What is JSP?

JSP which is the acronym for **Java Server Pages** is a server-side web programming technology whose major feature is to allow web developers to create platform-independent and dynamic methods for building Web applications. JSP has complete access to the whole family of Java APIs, which includes the JDBC API that gives access-to-database facilities.

A **JSP web page** is a text document that contains two kinds of text which are, static text which can be represented in any text-based format such as HTML, WML, XML, and XSL, and JSP elements which generate dynamic web content.

The **supported and recommended file extension** for a complete source file of a JSP page is **.jsp**. The page can be made up of a top file that can be also made up of other files that contain either a complete JSP page or pieces of a JSP page. The recommended extension for the source file of a piece or fragment of a JSP page is **.jspx**.



Features Of JSP

There are lots of features of JSP and some of them are:

- 1. Make interactive websites** One of the major uses of JSP is to develop very interactive webpages. Webpages that are not just static but are able to change content, dynamic and can communicate with the user in real time.
- 2. Easier to read data from the users** With JSP a user interacting with various contents such as textboxes, dropdown list, button, a checkbox can input some information. With JSP it is easier to accept that information entered by the user and forward it to the server.
- 3. Easier to display server response** After the data from the client's input is read and sent to the server, a response object is sent back to the client and it is easily displayed. For example, after a user fills and submits a form, a response from the server is displayed using JSP such as • "Congratulations, you have successfully registered for the event".
- 4. Allows you to add Java to your website** As you know JSP is the way to let Java Server Pages, one of the main features of JSP is to support Java code to be added inside your HTML code to give your webpage the programming power of Java and make it very powerful and interactive. JSP pages are internally converted into Java bytecode files, therefore all the features of Java, like security, platform-independence and flexibility is available in the JSP web pages as well.
- 5. Easier to connect to the database** One of the major features of JSP is that it easily allows us to connect our website with the database so that we can transfer the data that is stored by our user to a database and read it back from the database when it is required.
- 6. Easy to make JSP file** It is really about including Java to your webpage's HTML and it is equally straightforward and easy with anyone that has relevant knowledge of Java and HTML can easily learn and create JSP in a very short period.



Benefits Of JSP

There are lots of benefits of JSP and some of them are:

1. JSP webpages easily integrate with templates, including pieces of HTML or XML, with code that produces dynamic content.
2. JSP webpages are dynamically compiled into servlets when they are requested, so the page authors can easily create updates to the presentation code. JSP webpages can also be precompiled if you desire so.
3. The use of JSP for breaking Java/Servlet components handles these components internally, blocking the page author from the complexity of handling the application logic.
4. Developers can present well customized JSP tag libraries that other page authors can access making use of syntax like XML.
5. Web authors can edit and change the fixed template portions of web pages without directly affecting the software application logic.
6. Similarly, software developers can make changes to the application logic at the component level without modifying the individual web pages that use this logic.

Why Use JSP??

1. JavaServer Pages allow work with the same purpose as programs that are implemented using the Common Gateway Interface (CGI).
2. Performance is highly better due to the fact that JSP allows and supports the embedding of Dynamic Elements in HTML Pages itself instead of having separate CGI files.



JSP Course Outline

JSP • Introduction Overview
JSP • Environment Setup
JSP • Architecture
JSP • Lifecycle
JSP • Syntax
JSP • Directives
JSP • Actions
JSP • Implicit Objects
JSP • Content Request
JSP • Server Response
JSP • Request Objects
JSP • Form Processing
JSP • Writing Filters
JSP • Cookies Handling
JSP • Session Tracking
JSP • File Uploading
JSP • Handling File
JSP • Page Includes
JSP • Misc Content
JSP • Java Servlet
JSP • Sending Email
JSP • Standard Tag Library
JSP • Database Access
JSP • XML Data
JSP • Java Beans
JSP • Custom Tags
JSP • Expression Language
JSP • Exception Handling
JSP • Logging
JSP • Security
JSP • Internationalization
JSP • Video Lectures
JSP • Exams and Certification





BISMARCK
UNIVERSITY
Bismarck, North Dakota • Bismarck, North Dakota

ECLIPSE COURSE

DURATION: 2 WEEKS

FORMAT: WEBPDF PLUS VIDEO LECTURES

CERTIFICATE OF COMPLETION

What is Eclipse?

Eclipse is an integrated development environment (IDE) for Java programming language and other software development languages such as C, C++, Python, JavaScript, etc. The development environment that is provided by Eclipse includes the Eclipse Java Development Tools (JDT) for Java, Eclipse CDT for C/C++, and Eclipse PDT for PHP development, among many others.

The Eclipse development platform which provides the basic support for the Eclipse IDE is made up of plug-ins which is designed to be modular and extensible by making use of various additional plug-ins. Eclipse platform which was developed using the Java programming language is used to create rich-client applications, integrated software development environments, and other tools. Eclipse can be used as a development tool for any programming language for which a plug-in is already available.

The Java Development Tools (JDT) project provides developers with a plug-in that allows Eclipse to be used as a Java IDE. Python is an Eclipse plugin that allows Eclipse to be used as an IDE for Python projects. C/C++ Development Tools (CDT) is an Eclipse plug-in that lets Eclipse to be used for developing C/C++ applications and software.

The initial publisher of Eclipse started from IBM Monksage. The Eclipse software development kit (SDK), which comes with various Java development tools, is meant for Java developers. Users can also increase their abilities by installing the various plug-ins that are writers for the Eclipse Platform, such as development utilities for other programming languages, and can develop and contribute their own plug-in modules.

The Eclipse SDK (Eclipse IDE Development Kit) is a free and open-source software application that is released under the terms of the Eclipse Public License, although it is not compatible with the GNU General Public License. It was one of the first Integrated Development Environment to run under the GNU C library and it runs without problems under *linux* too.



History of Eclipse

Eclipse was motivated by the VisualAge genealogy of integrated development systems, more IDE software. Although it was fully successful, the main problem of the VisualAge software was that the entire code was not in a module-based software model. Instead, all of the code for a software project was kept in a flattened heap. Single classes weren't really removed, definitely not outside the IDE.

A group at the IBM Cary, NC laboratory, made the first software as a free-based system, more in 2001 around the month of November, announcement was created with a board of attendees to push the creation of Eclipse as an open-source product.

It is said that IBM initially invested close to \$10 million at that time. The founding members were: Borland, IBM, Microsoft, QNX Software Systems, Rational Software, Red Hat, Sun, TogetherSoft, and Webtrends.

The number of attendees grew to 40 at the close of 2001. In 2004 around January, the Eclipse Foundation was formed.

Features Of Eclipse

- 1. Software Updates:** Eclipse provides an update site, where you can go and get the latest version of a framework or plugin, this feature makes the developer's life really easy. You never need to worry about dependencies, or installing things, it is done for the plugin itself. Everything is completely handled for you through a simple dialog.
- 2. Enterprise Java Tooling:** Eclipse has some of the best tooling available for EE projects. The amount of functionality available in Eclipse is outstanding, from basic wizard for the creation of Web Services, standard EE and EJB tooling, up to GQL editing.
- 3. Model-Driven Development:** The Eclipse Modelling Project presents you with a nice set of modeling tools for those using EMF and other related modeling technologies. The Model Tools that are available to work with EMF models allows you to create and modify your overview a standard way or using the visual flow diagram editor.



Benefits Of Eclipse

1. Using the Eclipse IDE will save you less time and effort.
2. Navigating files with Eclipse IDE is made very easy.
3. Auto-completion in Eclipse is one of the best features, you don't have to remember all the syntax of your language.
4. Refactoring
5. Error debugging is easy you can easily navigate to the Error line.
6. Learning Java and using Eclipse concept gives you a lot of job opportunities with high pay.



The **Modeling system** encompasses all the official software projects of the Eclipse Foundation, that mainly based on model-transformation systems. All are related with the Eclipse Modeling Framework that was created by IBM.

Model Transformation software projects make use of Eclipse Modeling Framework (EMF), based models, as a generic and generative model or even as an output. Model to model conversion projects includes ATL (ATL Transformation Language, a unique open-source conversion language and toolflow that is suitable to create a specific model or to create a very rich model from a specific EMF model. Model to text conversion software projects consist of Acceleo, an maker of M2TEXT, suitable model to text language from the Object Man. approach Group. The iterative code creator can generate any textual language (Python, PHP, Java, etc.) from EMF modelled models combined with any metamodel (UML-like Modeling Language, Systems Modeling Language, etc.)

Model Development Tools software projects are made of different modeling specific libraries used in the industry and their toolflows. Among these projects can be found makers of various standards:

1. Unified Modeling Language (UML);
2. Systems Modeling Language (SysML);
3. Object Constraint Language (OCL);
4. Business Process Model and Notation (BPMN);
5. Interactive Media Manager (IMM);
6. The semantics of Business Vocabulary and Business Rules (SBVR);
7. XML Schema (XSD);
8. National Electronic Distributors Association (NECA).

The **Concrete System Development** software project embodies the Graphical Modeling Framework, an Eclipse-based framework made for the graphical representation of UML-based models.

The **Abstract System Development** software project borrows the Eclipse Modeling Framework, the base of most of the modeling software projects of the Eclipse Foundation inside framework that is available.

Technology and Research software projects are the special protection of the Modeling Group, this project is used to know all the modeling software projects of the Eclipse Foundation during their development phase.

Amalgam provides the packaging and combination among all the available modeling tools for the Eclipse package especially made for modeling tools.



Why Study Eclipse?

1. Knowledge of Eclipse as a Java Developer can get you a lot of job opportunities with high pay.
2. Eclipse is an Object-Oriented Language
3. Navigating through your code files with Eclipse IDE is made easy, together with many tools for auto-completion, debugging and refactoring.
4. Job Opportunities and Career Advancement.



Eclipse Course Outline

Eclipse • Introduction
Eclipse • Installation
Eclipse • Explore Windows
Eclipse • Explore Menus
Eclipse • Explore Views
Eclipse • Perspectives
Eclipse • Workspaces
Eclipse • Create Java Project
Eclipse • Create Java Package
Eclipse • Create Java Class
Eclipse • Create Java Interface
Eclipse • Create JAR File
Eclipse • Java Build Path
Eclipse • Run Configuration
Eclipse • Running Program
Eclipse • Create Test File
Eclipse • Clone Project
Eclipse • Import Project
Eclipse • Build Project
Eclipse • Debug Configuration
Eclipse • Debugging Program
Eclipse • Performance
Eclipse • Content Assist
Eclipse • Quick Fix
Eclipse • Hover Help
Eclipse • Search Menu
Eclipse • Navigation
Eclipse • Refactoring
Eclipse • Add Bookmarks
Eclipse • Task Management
Eclipse • Install Plugins
Eclipse • Code Templates
Eclipse • Shortcuts
Eclipse • Window Editor
Eclipse • Tips & Tricks
Eclipse • Web Resources
Eclipse • Video Lectures Eclipse • Exams and Certification





BISMARCK
UNIVERSITY
The Learning Never Ends

APACHE HTTP SERVER COURSE

DURATION: 2 WEEKS

FORMAT: WEBPDF PLUS VIDEO LECTURES

CERTIFICATE OF COMPLETION

What is Apache?

Apache is the full Apache HTTP server is a powerful and robust web server software. It is open source and free to download and use. Apache is considered by many as a leading tool in the world of webserver software. It is also the most widely used open software on the web.



Advantages of Apache HTTP Server

1. Running Apache server leaves you with less problems whether it speaks to a pure platform (web software) i.e. it can run on any operating system, be it Windows, Linux, Mac OS, or the emerging IoT.
 2. Apache is known to have made helpful contributions to the growth of the World Wide Web (www) and is still adding more value to it, this remarkable contribution have given it a lead in web server software market, and its developers are working continuously to make it even much better and flexible.
 3. Some of the programming language supported by Apache include Python, PHP, Perl, and others.
 4. Apache has a lot of features (support) many of which are implemented as an extension of the core functionality of compiled modules.
 5. Apache also has support for authentication modules (examples of these modules include: mod_auth, mod_auth_proxy, and mod_auth_digest).
 6. Some other features supported by Apache include cookie negotiation, proxied authentication, compressed user interface, PFB, DCA, digital certificates, to mention but a few.
 7. Apache runs with the primary name httpd. This means multiple websites can run on a single Apache installation, the unique feature is made possible by what is known as Virtual hosting. Virtual hosting is a way of hosting several domain names on a single server.
 8. Apache also gives an inherent implementation a lot of flexibility (through what is known as Multi-processing Module), the choice of Multi-processing Module is of great importance as it allows Apache to run in a hybrid, process based mode. One of the major drawbacks of Apache 2.2 was its sluggishness in the delivery of static file, all thanks to the developers that immediately ran to the rescue with Apache 2.4 which of course ran to the rescue back.
 9. Apache is still being actively marketed and will continue to be a preferred choice amongst its competitors.
10. Apache is available free-of-cost that requires no license.
 11. Apache is reliable and high performance.
 12. Apache is easy to install and easy to configure/web server.
 13. Apache efficiently support and faster communication.
 14. Apache offers instant changes, so to modify content the server file changes is reflect.
- In the Full course, you will learn every thing you need to know about Apache HTTP Server with 5 certification upon successful completion of the course.**



Apache Content Outline:

- Apache • What is Apache?
- Apache • How to install Apache
- Apache • Install Apache on Linux Platform
- Apache • Install Apache from Source
- Apache • What is Virtual Host?
- Apache • Types of Apache Virtualhost
- Apache • Name-based Virtual Host
- Apache • IP-based Virtual host
- Apache • What Apache needs to Run/Log File?
- Apache • PID/Processes in Apache
- Apache • How to run Ruby with Apache
- Apache • How to Secure Apache Web Server?
- Apache • Hiding Apache version and OS information
- Apache • Disable Directory Listing
- Apache • Disabling unnecessary modules
- Apache • Restricting Access to Files outside the web root directory
- Apache • Using mod_security to enhance the DDoS attack
- Apache • Using mod_security to enhance Apache security
- Apache • Limiting request rate
- Apache • Apache Log Format
- Apache • Available Apache Directives
- Apache • Types of Apache Log Format
- Apache • Common Log Format
- Apache • Combined Log Format
- Apache • Configure your very first Production Web Server
- Apache • Exams and Certification





BISMARCK
LEADERSHIP TRAINING & DEVELOPMENT
Helping Leaders Become Leaders. Every Day. Every Where.

JYTHON COURSE

DURATION: 2 WEEKS

FORMAT: WEBPDF PLUS VIDEO LECTURES

CERTIFICATE OF COMPLETION

What is Python?

Python is the first Virtual Machine implementation of the Python programming language. Python was developed to run on the Java platform. A Python software program can import arbitrary classes that are in Java. And in Java, Python source code compiles to Java byte code. One of the main benefits of Python is that a more interface that is designed in Python can make use of GUI elements of SWT, Swing or SWT Plugins.

The **Python project** was started as Python and later the name was changed to Jython. It follows closely the standard Python implementation which is known as CPython as developed by Guido van Rossum who is the developer of Python. Python was developed in 1991 by the creator, Python 2.0 was later released in 1999. Since then, the Jython 2.x release series corresponds to the equivalent CPython releases. Python 2.7.0 was released in May 2010, and it corresponds to CPython 2.7. The development of Jython 2.x is currently under progress.

Features of Jython

There are many features of Jython and some of them are

- 1. The Jython Registry:** Jython runs with a local Jython registry file which is used to provide a platform-independent equivalent to the Windows registry file. It joins this together with environment variables and system file information at startup.
- 2. Embedding:** With Jython, Java classes can be embedded into Python scripts, and the Python scripts can be loaded and imported from the Java code.
- 3. Collections and Array support:** Jython provides you with many object-oriented interfaces, Java arrays and collections with Python data structures.
- 4. Compiling to Java class files:** The compile all modules that is present in Jython (Jython produces Java byte code in Java class files from Python code). It currently corresponds to the CPython module of the same name. (Earlier versions make use of a tool that is known as Jythonc which was discarded fully in Jython 2.2).
- 5. Database interaction:** The JJCDBC module provides you with a Pythonic API interface to support the Java Database Connectivity (JDBC API).



Benefits of Python

There are lots of features of Python and some of them are:

- 1. Readability of Source Codes:** The usability of Java classes and interfaces in your system code with Python is one of the best deals that are available. Also, other features that make Java stand out from the crowd, such as automatic garbage-collection and multithreading, enhance the feature set of Python programs.
- 2. Java bytecode compilation:** One of the biggest strengths of Java that motivated the programming community when it was first introduced was the concept of the development of bytecode. In the form of a *.class file, several pieces of microprocessor-like code is generated for every class and interface that is present inside your Java file. Apart from making a head in making the language to be portable across various platforms, it also improves the performance of the applications as well.
- 3. Easy-to-use runtime:** Python simplifies the process of creating classes.
- 4. High-Level Language Abstraction:** Python gives you a very nice combination of High-Level Language Abstraction and the most popular scripting language. It is also great if you need to implement scripts throughout your business, as if you just need a small piece of code or a method to initiate a query in a database.

Differences between Python and Jython

The below are the differences between Python and Jython:

- 1. The reference implementation of Python, called CPython, is written in C language. While Python, is completely written in Java and is a Java Virtual Machine implementation.**
- 2. Standard Python is available on multiple platforms. Python is available for any platform with a Java Virtual Machine installed on it.**
- 3. Standard Python code compiles to a .pyc file, while the Jython program compiles to a .class file.**
- 4. Python extensions can be written in C language. Extensions for Jython are written in Java.**
- 5. Python is truly multi-threaded in nature. While Python makes use of the Global Interpreter Lock (GIL) mechanism for the process.**
- 6. Both implementations have different garbage-collection mechanisms.**





Mathias Bevil, St. Johns College, Florida, United States
E-mail: mathias@stjohnscollege.edu
www.stjohnscollege.edu
E-mail: mathias@stjohnscollege.edu