Bright Star Institute

302.335.8868

admissions@brightstarinstitute.com

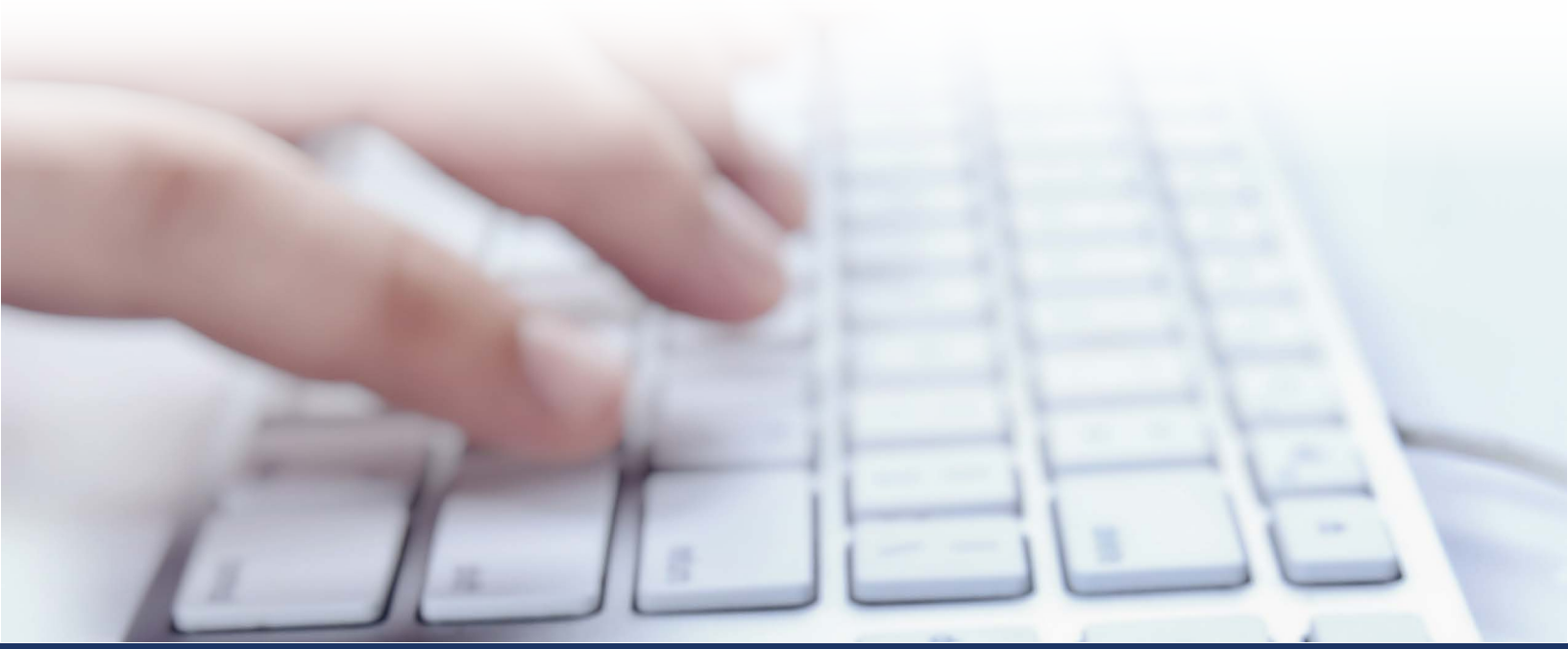
Java Developer Certificate

Online, self-paced training that is focused on giving you the skills needed to stand out.

Online learning gives you the flexibility to plan study time around the rest of your day. Our programs combine video, hands-on lab exercises and one-on-one facilitation to enable you to learn as your schedule allows.

Labs are performed via remote access to a dedicated lab machine (unless a student requests to utilize their own equipment). Our unique commitment to student success includes a personal instructor to assure that your courses are completed on time and with full understanding of both the lecture content and the lab exercises.

Course Advisors are available at 302.335.8868 to answer any questions you may have.



Java Developer Certificate Overview

4-course program: $4,500

The Java Developer certificate program focuses on the skills needed to code, compile and run standalone object-oriented Java programs. Object-oriented concepts and proper use of classes, objects, inheritance and polymorphism are emphasized throughout the training. Students will learn how to make sophisticated data queries using SQL (Structured Query Language) and will be able to interact with SQL databases from their Java applications via Java Database Connectivity (JDBC). By the end of the course, students will also be proficient with the Spring framework, and will be able to leverage its powerful capabilities when developing Java web applications.

Required Courses:

Course Name	Duration
Java Programming	90 Hours
SQL Programming	60 Hours
Introduction to Spring 5, Spring MVC and Spring REST	90 Hours
Effectively Using Java Packages And Features	90 Hours

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Optional Courses:

Course Name	Duration
Introduction to Programming Concepts	15 hours
Learning to Program with Java	90 hours
Website Development with HTML5, CSS and Bootstrap	90 hours
Test Driven Development (TDD), and Refactoring Legacy Code Using Java	75 hours

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Required Courses

Java Programming

Duration: 90 hours

Prerequisites: Basic programming skills in a structured language. Knowledge and experience with Object-Oriented Design (OOD) is helpful, but not required.

Course Description: This hands on course introduces experienced programmers to Java™ technology and Java programming techniques. The Java platform provides an object-oriented, portable and robust framework for application development. Included are core language concepts including fundamental data types, flow control, and standard function libraries. The course emphasizes object oriented programming and modular design to support distributed development environments. Included are the design of classes and objects, inheritance and polymorphism, and the details about creating programs for use on a distributed network, with emphasis on JSP, Servlets, and JDBC. The course also includes coverage of the Java Collections API, fundamental I/O, exceptions, and exception handling.

The course is designed to leverage the participants' existing programming skills and to highlight the new and extended features of the Java programming framework as compared to other common languages. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students Will Learn:

- Introduction to Java
- Developing Software With Java SE
- Java Fundamentals
- Flow of Control
- Data Types and Operators
- Using Java Objects
- Arrays
- Developing Java Classes
- Inheritance and Interfaces
- Type Safety
- The Collections Framework
- Exceptions and Exception Handling
- Basic Input and Output (I/O
- Network Programming
- Threads
- Java Database Connectivity (JDBC)
- Introduction to Java Web Applications
- Rich Internet Applications with Java

SQL Programming

Duration: 60 hours

Prerequisites: General computer knowledge.

Course Description: This SQL programming course teaches students relational database fundamentals and SQL programming skills. Topics covered include relational database architecture, database design techniques, and simple and complex query skills. This class is intended for analysts, developers, designers, administrators, and managers new to the SQL programming language. Upon completion, participants will understand SQL functions, join techniques, database objects and constraints, and will be able to write useful SELECT, INSERT, UPDATE and DELETE statements.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students Will Learn:

- Introduction to Relational Databases and SQL
- Designing Relational Databases
- Creating Databases and Tables
- Working with Records
- JOIN Statements
- Advanced SELECT Statements
- Understanding Subqueries
- SQL Procedural Programming
- Views and Triggers
- Database Security and Transactions

Introduction to Spring 5, Spring MVC and Spring REST

Duration: 90 hours

Prerequisites: Java SE programming experience and an understanding of object-oriented design principles. Fundamental knowledge of XML is helpful but not required. The course **Java Programming** or equivalent knowledge provides a solid foundation.

Course Description: Spring 5 provides an evolutionary advance of Spring's powerful capabilities. This course introduces those capabilities, as well as providing guidelines on when and how to use them. It includes coverage of the three main configuration styles: Java-based (@Configuration), annotation-based (@Component), and the traditional XML-based configuration that may still play an important role in existing and new projects.

The course starts with in-depth coverage of Spring's Core module to reduce coupling and increase the flexibility, ease of maintenance, and testing of your applications. It goes on to cover many of the most important capabilities of Spring, including easing configuration with Spring Boot, integrating Hibernate and JPA persistence layers with Spring and Spring Data, and using Spring's declarative transaction capabilities.

The course includes integration of Spring with Java EE Web applications, a solid introduction to Spring MVC, and coverage of building RESTful resources with Spring MVC. It also provides an overview of Springs reactive programming model for repositories and Web resources.

Students Will Learn:

- Introduction to Spring
- The Spring Container
- Dependencies and Dependency Injection
- Annotation-based Configuration
- Java-based Configuration
- Integrating Configuration Types
- Bean Scope and Lifecycle
- Value Injection
- Constructor Injection
- Qualifiers / DSL
- Database Access With Spring
- Spring AOP Introduction
- Spring Transaction Management
- Java EE Web Applications with Spring
- Spring MVC Basics
- View Resolvers
- Controller Details
- Forms and Model Objects
- Working with Sessions
- Validation
- RESTful Services with Spring
- Common REST Patterns

Effectively Using Java Packages And Features

Duration: 90 hours

Prerequisites: Java SE programming experience and an understanding of object-oriented design principles. Fundamental knowledge of XML is helpful but not required. Bright Star's **Java Programming** course or equivalent knowledge provides a solid foundation.

Course Description: This intermediate level course is intended for programmers who already have a fundamental understanding of Java programming and some experience writing code. It provides additional insights and details regarding some of the more advanced and useful capabilities contained in the Java Programming Language and its associated packages. Topics include reflection and JavaBeans, Java type safety enhancements, the Java Collections Framework, Java Database Connectivity (JDBC), multithreading, inner classes, lambda expressions and networking.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students Will Learn:

- Reflection Using the Class Object
- Invoking Methods Using Reflection
- Methods Available on the Class Object
- Customizing BeanInfo
- Type Safety Enhancements
- Collections Framework
- Simple Arrays and Arrays of Objects
- Legacy Container Classes
- Data Sharing Among Threads
- Coordination and Controlling Threads
- Synchronizing Threads
- Working with JDBC
- Overview of java.net Package
- Inner and Nested Classes
- Lambda Expressions

Optional Courses

Introduction to Programming Concepts

Duration: 15 hours

Prerequisites: Rudimentary knowledge of computer systems. Students should know how to basically operate a computer, browse the world wide web, send and receive email messages, and write and print memos.

Course Description: Courses that teach programming languages tend to focus on the particulars of each language (as they should), expending only minimal time discussing basic universal programming concepts. Students who are new to programming may find themselves overwhelmed in discussions of the details of matters such as control flow operators, executable images, third generation languages, and logical operators before they really understand the concepts. This situation can leave the student lost or frustrated or both.

This course is offered for the aspiring programmer to provide them the foundational knowledge that is necessary to learn and advance in a programming career. Students will be exposed to fundamental concepts that are utilized by all programming languages in this course.

Students Will Learn:

- Basic computer terminology and jargon
- Programming tasks: Analysis and requirements gathering
- Programming tasks: Design and planning
- Programming tasks: Coding
- Programming tasks: Testing
- Programming tasks: Deployment/ Documentation
- Programming tasks: Maintenance
- The nature, pros and cons of compiled vs. interpreted languages
- The steps of compiling: parsing, optimization, object code, linking, executable images
- How to write and compile a rudimentary C program
- Programming styles: imperative, structured, procedural, object-oriented, declarative, and functional
- The utility of stand-alone, distributed, client-server and web-enabled programming
- Statements, commands, comments and reserved words
- Routines, subroutines, functions and libraries
- Control flow structures
- Algorithms

Learning to Program with Java

Duration: 90 hours

Prerequisites: Familiarity with computers.

Course Description: This hands on basic Java training course provides an introduction to programming using the Java language. Students are introduced to the application development cycle, structure of programs, and specific language syntax. The course introduces important algorithmic constructs, string and character manipulation, dynamic memory allocation, standard I/O, and fundamental object-oriented programming concepts. The course explains the use of inheritance and polymorphism early on so the students can practice extensively in the hands on labs. Structured programming techniques and error handling are emphasized. The course includes the processing of command line arguments and environment variables so students will be able to write flexible, user-friendly programs. Comprehensive hands on exercises are integrated throughout to reinforce learning Java programming and develop real competency.

Students Will Learn:

- Fundamental elements of programming
- Interactive Development Environment (IDE) concepts
- Classes, objects and methods
- Declaring and instantiating a Java object
- Using conditional and looping constructs
- Declaring and instantiating arrays
- Using and creating interfaces
- Defining classes using inheritance
- Exception handling
- Formatting output with class Formatter
- Using strings, characters and regular expressions
- GUI programming concepts

Website Development with HTML5, CSS and Bootstrap

Duration: 90 hours

Prerequisites: Basic personal computer skills and basic Internet knowledge.

Course Description: This hands on course provides a thorough introduction into the creation of a Website using HTML, CSS and Bootstrap. The course starts with thorough coverage of HTML and Cascading Style Sheets (CSS) and progresses to using the Bootstrap framework to create mobile-friendly websites.

Topics include use of HTML5 semantic tags, block-level and inline elements, creating links, ordered and unordered lists, creation of tables and forms. Students will learn to attach CSS to a page using several different techniques, CSS selectors and pseudo-classes, CSS box model, and a variety of CSS properties.

Students will learn how to create a Bootstrap page utilizing the grid system, implement commonly used components (such as dropdowns and navigation bars), use Bootstraps CSS classes to format page elements and use Bootstrap's plugins to add tabs, modals and accordions to a page.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students Will Learn:

- Creating a valid HTML document
- Using various tags to markup text
- Creating and using CSS to format the appearance of an HTML document
- Using HTML5 semantic elements to define the structure of a page
- Changing the appearance of elements in a document with type, id, class and attribute selectors
- Creating CSS3 effects such as drop shadows, text shadows and using custom fonts
- Creating ordered, unordered and description lists
- Displaying images on a Web page
- Creating hyperlinks using text or images
- Using the a tag to create anchors in a page
- Displaying information using an HTML table
- Including forms in an HTML document
- Adding video and audio to a Web page
- Building a Web page using the Bootstrap grid system
- Using Bootstrap to style links, buttons and form controls
- Adding components to a page such as dropdowns, thumbnails, alerts and glyphicons

Test Driven Development (TDD), and Refactoring Legacy Code Using Java

Duration: 75 hours

Prerequisites: Java programming experience and an understanding of object-oriented design principles.

Course Description: This course provides students with hands on experience learning Test Driven Development (TDD) using JUnit. Students will build unit tests using mocks, fakes, stubs and drivers, and address issues working with databases and other systems. Student will create tests and code that will be more likely to meet and exceed requirements. Code that receives "test coverage" will not break existing systems, because tests are passed before code is checked in.

Students will spend time working with the issues involved in refactoring legacy code, safely cutting into an already deployed system. Students will work on looking for, or creating "seams" to more safely improve code or add features, and work on identifying "code smells" that need attention in a productive system. Finally, students will explore dependency issues as well as techniques to better understand and improve complex systems.

Students will also examine TDD and refactoring legacy code in other languages like C# to gain a broader view of options and issues working in a multi-language shop. Comprehensive labs using Java provide facilitated hands on practice crucial to developing competence and confidence with the new skills being learned.

Students Will Learn:

- Why Test?
- Unit Testing Basics
- Unit Testing Details
- Mocks, Fakes, Stubs and Drivers
- Database Testing
- Refactoring Basics
- Patterns and Anti-Patterns in Test Driven Development
- Refactoring Legacy Code
- Analysis and Coverage
- System, Regression and User Acceptance Testing