Bright Star
Institute

☎ 978.256.9077

✉ admissions@brightstarinstitute.com

Learning to Program with Java

Duration: 35 hours

Price: $900

Prerequisites: Familiarity with computers.

Description: This hands on Java Programming 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 and develop real competency.

Course Overview

Fundamentals of the Program Development Cycle

- Computer Architecture
- The Notion of Algorithms
- Source Code vs. Machine Code
- Compile-Time vs. Run-Time
- Software Program Architecture
  - Standalone
  - Client/Server
  - Distributed
  - Web-Enabled
- IDE (Interactive Development Environment)

Application Development Fundamentals

- Structure of a Java Program
- Memory Concepts
- Fundamental Data Type Declarations
- Fundamental I/O Concepts
- Fundamental Operators
  - Arithmetic Operators
  - Logical Operators
  - Precedence and Associativity
- Building and Deploying a Java Program

Concepts

Introduction to Classes and Objects

- Classes, Objects and Methods
- Object Instances
- Declaring and Instantiating a Java Object
- Declaring Methods
- `set` and `get` Methods
- Initiating Objects with Constructors
- Primitive Types vs. Reference Types

Writing Methods (Functions)

- Static vs. Dynamic Allocation
- Declaring Methods
- Declaring Methods with Multiple Parameters
- Method-Call Stack
- Scope of Declarations
- Argument Promotion and Casting
- Designing Methods for Reusability
- Method Overloading

Deeper Into Classes and Objects

- Controlling Access to Class Members
- Referencing the Current Object Using `this`
- Overloading Constructors
- Default and No-Argument Constructors
- Composition of Classes
- Garbage Collection and Destructors
- The `finalize` Method
- Static Class Members

Increasing Convenience by Using Polymorphism

- Purpose of Polymorphic Behavior
- The Concept of a Signature
- Abstract Classes and Methods
- `final` Methods and Classes
- Purpose of Interfaces
- Using and Creating Interfaces
- Common Interfaces of the Java API

Flow Control

- Conditional Constructs
- Looping Constructs
    - Counter-Controlled Repetition
    - Sentinel-Controlled Repetition
- Nested Control Constructs
- `break` and `continue` Statements
- Structured Programming Best Practices

Arrays

- Purpose of Arrays
- Declaring and Instantiating Arrays
- Passing Arrays to Methods
- Multidimensional Arrays
- Variable-Length Argument Lists
- Using Command-Line Arguments
- Using Environment Variables

Defining Classes Using Inheritance

- Superclasses and Subclasses
- Advantages of Using Inheritance
- `protected` Class Members
- Constructors in Subclasses

Files and Streams

- Concept of a Stream
- Class File
- Sequential Access
- Object Serialization to/from Sequential Access Files
- Additional java.io Classes

Fundamental Searching and Sorting

- Introduction to Searching Algorithms
  - Linear Search
  - Binary Search
- Introduction to Sorting Algorithms
  - Selection Sort
  - Insertion Sort
  - Merge Sort

Exception Handling

- Types of Exceptions
- Exception Handling Overview
- Exception Class Hierarchy
- Extending Exception Classes
- When to Throw or Assert Exceptions

Strings, Characters and Regular Expressions

- Fundamentals of Characters and Strings
- `String` Class
- String Operations
- `StringBuilder` Class
- `Character` Class
- `StringTokenizer` Class
- Regular Expressions
  - Regular Expression Syntax
  - `Pattern` Class
  - `Matcher` Class

Fundamental Data Structures

- Dynamic Memory Allocation
- Linked Lists
- Stacks
- Queues
- Trees

Formatted Output

- `printf` Syntax
- Conversion Characters
- Specifying Field Width and Precision
- Using Flags to Alter Appearance
- Printing Literals and Escape Sequences
- Formatting Output with Class `Formatter`

Fundamental GUI Programming Concepts

- Overview of Swing Components
- Displaying Text and Graphics in a Window
- Event Handling with Nested Classes
- GUI Event Types and Listener Interfaces
- Mouse Event Handling
- Layout Managers

978.256.9077

admissions@brightstarinstitute.com