



## Introduction to Programming – Java

Introduction to Programming – Java is broken into two parts. Part 1 covers units 1 to 5, and part 2 covers units 6 to 10. Each unit concludes with a unit test and assignment. At the end of each part there is a final project and final exam.

### Course Description:

Introduction to Programming Java introduces students to computer science concepts and basic programming techniques. In this course, the students learn:

- The fundamentals of computer programming in Java
- The broader field of computer science and algorithmic thinking
- The fundamental approach that computer scientists take to solving problems.

Students will develop creative solutions for various types of problems as their understanding of the computing environment grows. In this course, we will also introduce the Java Programming Language to the students as their chosen technology in solving simple to intermediate coding problems.

### Unit 1: Program Design

Students begin their studies by learning what programming is and its importance in today's society. The unit covers the various types of programming languages, what they are and their functions in programs. Students learn about the Java development environment.

#### Unit 1 Modules:

- How to setup the Java Development Kit (JDK) and an Integrated Development Environment (IDE) to start coding
- The Importance of Programming
- Types of Programming Languages
- What is Java?
- How Java Code is Processed by the Computer
- Etiquettes in Programming

#### Unit 1 Learning Objectives:

- Students learn what programming is, how it works and its importance.
- Students are introduced to algorithms and simple data structure.
- Students learn about individuals who made a significant change in our society by being a coder.
- Students study the different classifications of programming languages and their use.
- Students learn how to define the various types of programming languages, understand their functions and roles in computer science and give examples of each.

- Students learn how to set up a development environment, what a development environment is and how a development environment works.

## Unit 2: Number Systems and Logic

Unit 2 introduces students to the many different type of number systems. Students learn the importance of the number system as related to computer programming. Students learn the Binary system and the role of 0's and 1's in programming and how to read sequences of numbers based on different number systems (i.e: binary, decimal, octal and hexadecimal). This unit also covers flow charts and logic formulation, as well as the order of operation to solve basic to complex numeric equations.

### Unit 2 Modules:

- The Decimal, Binary and Hexadecimal Number System
- Introduction to Binary
- The Hexadecimal Number System
- Introduction to Octal
- Number System Conversion
- Signed and Unsigned Numbers
- Process Flow Charts
- Order of Operation

### Unit 2 Learning Objectives:

- Students learn the importance of the different types of number system in the computer systems. Including the common types of number systems: Binary, Decimal, Octal and Hexadecimal.
- Students learn how to convert from one number system to another. This lesson allows students to use their analytical and critical thinking skills.
- Students study the importance of 'signed' and 'unsigned' numerical values.
- Students are introduced to flow charts and learn how to create and use a process flow chart in graphically depicting a programming logic.
- Students study the order of operations to solve complex numeric equation that involves several operations.

## Unit 3: Basic Java Syntax

This unit guides the students through the complete set up of the Java Development Kit (JDK) and the Eclipse Integrated Development Environment (IDE) on their computer. Students also create their first Java application.

### Unit 3 Modules:

- Setting the Java Development Kit (JDK), Eclipse IDE and The HelloWorld Program Page
- Introduction to jshell

- Java Program Structure
- Keywords and Identifiers
- Variables
- Data Types - Primitive Type and Reference Type
- Strings
- Methods in Java
- Passing Simple Parameters in Java Methods
- Commenting - Single-line, Multi-line and JavaDoc

#### Unit 3 Learning Objectives:

- Students learn the fundamentals of coding and the basic Java syntax.
- Students learn to code their very first Java Program: the "Hello World" Program.
- Students learn how to pass parameters, as well as how to invoke keywords and identifiers in their code.
- Students learn the concept of variables, Strings, and Methods and how to implement them.
- Students learn to compile and run a Java source file and debug runtime errors.
- Students are introduced to important practice of adding comments in their code.

#### Unit 4: Operators

This unit focuses on the different type of operators. It builds on prior lessons so that students continue to add to their knowledge base and critical thinking skills by studying operations as related to programming and computer science. Students study traversal and logical operators. This gives students a better understanding of the operational aspect of programming.

#### Unit 4 Modules:

- Casting Operator
- Arithmetic Operators
- Relational Operators
- Logical and Short Circuit Operators
- Assignment and Short-Hand Operators
- Ternary Operator

#### Unit 4 Learning Objectives:

- Students learn how to change the data type of a primitive variable using casting operators.
- Students learn how to implement the mathematical operators in Java that are composed of unary operators and the arithmetic operators.
- Students study the use of logical and relational operators for condition checking.
- Students study the use of short-circuit, ternary, assignment and the short-hand operators for solving programming problems.

- Students learn how to define and give examples of these various operators, understand their structures and write computer programs using them.

## Unit 5: Control Structures

This unit concentrates on control structures as it relates to computer science and programming. Students learn that all programming languages have some form of “if” statements that allows them to test conditions. Students learn about various programming structures including conditionals, iteration and recursion. Students learn to identify and define these structures and give examples of each.

### Unit 5 Modules:

- Control Flow Statements - If, If-else, if-else-if and Nested If statements
- The switch-case Statement
- The Concept Behind Control Loops
- The While Loop
- The do-while Loop
- For Loop
- The Break Statement
- The Continue
- Labeled Breaks and Labeled Continue Statements

### Unit 5 Learning Objectives:

- Students study and learn the concepts of flow control structures in Java.
- Students learn the variety of flow control statements in Java such as the IF-else Statement, Switch-Case Statement, the Do-While, the While and For Loop and how to implement them.
- Students learn the concepts of looping, and are introduced to the different types of loop constructs.
- Students learn to apply logical thinking by adding versatility to their programs.
- Students learn how to give decision-making abilities to their programs.

## Unit 6: Arrays

In unit 6 students’ study what an array is, how to create an array, and how to use an array. Students learn about the Command Line, accepting values via command line arguments and how to manipulate those values. This unit allows students to utilize prior lessons and use their critical thinking skills.

#### Unit 6 Modules:

- Single Dimensional Arrays
- The Command Line Arguments
- Two-Dimensional Arrays
- Non-Rectangular, Two Dimensional Arrays

#### Unit 6 Learning Objectives:

- Students learn what arrays are and how to declare, create and initialize an array object, as well as how to get the array limits.
- Students learn how to copy the elements of an array object to another array using the `System.arraycopy()` method.
- Students learn how to use the command line arguments.
- Students learn how to declare, create and use an array of arrays (two dimensional arrays).
- Students study the difference between a rectangular and non-rectangular two-dimensional array.

#### Unit 7: Error and Exceptions in Java

In this unit students begin their study into testing and debugging as it relates to the study of programming and computer science. Students learn about programming errors, what they are and how to find them. This unit allows students to utilize their critical thinking skills.

#### Unit 7 Modules:

- Overview on Java Programming Errors
- Debugging your Java Source Code
- The Error and Exception Objects
- Creating and Declaring Exceptions
- Assertion Checks (the `assert` keyword)

#### Unit 7 Learning Objectives:

- Students are exposed to common programming errors, like compile time (AKA Syntax Errors), logical and runtime errors.
- Students learn how identify, handle and debug errors using the integrated development environment (IDE).
- Students learn about the error and the exception objects including some of their sub types.
- Students learn about the use of the `assert` keyword and how it is commonly used for testing.

## Unit 8: Classes and Objects

This unit introduces classes and objects, it focuses on the difference between them and how they are used. Students learn how to leverage the use of object-oriented programming techniques in creating simple to intermediate applications.

### Unit 8 Modules:

- Introduction to Object-Oriented Programming Concepts in Java
- Classes & Objects
- Constructors and the Garbage Collector
- Encapsulation and Access Modifiers in Java
- Inheritance
- Overriding Methods
- Inheritance and Constructors (Constructor Chaining)
- Polymorphism
- The Static Keyword
- The toString(), equals() and hashCode() methods of the Object Class

### Unit 8 Learning Objectives:

- Students learn about Object Oriented Programming (OOP) concepts including Abstraction, Encapsulation, Inheritance and Polymorphism.
- Students learn the importance of the Java OOP concepts in creating working methods and variables, that can be re-used all or in part without compromising security.

## Unit 9: Advance Object-Oriented Programming (OOP) Concepts and Input/Output Statements

Unit 9 allows students to utilize prior lessons as they move on to more advanced concepts in OOP, including wrapper classes and essential keywords that are required to implement them. Students learn the concept behind input and output techniques on data entry and files.

### Unit 9 Modules:

- Wrapper Classes
- The enum Keyword
- The final Keyword
- The abstract Keyword
- Interfaces
- How to accept Inputs from the Keyboard using GUI Components
- The BufferedReader and InputStreamReader Classes
- The Scanner Class
- File I/O

### Unit 9 Learning Objectives:

- Students learn the concept behind wrapper classes and how to implement them.
- Students learn how to use the special Java type, the enum keyword, as well as the uses for the final and abstract keywords.
- Students are introduced to interfaces and how to use them in their applications.
- Students learn how to prompt users to enter information, create dialog windows and to accept inputs from the keyboard using graphical user interface components.
- Students learn how to use input and output files.

### Unit 10: Creating your Java Technology GUI Application

Unit 10 culminates in utilizing all prior lessons, enabling the student to write a Java graphical user interface application. Students learn the key principles behind GUI programming. Students learn how to apply event handling techniques to create desktop applications.

#### Unit 10 Modules:

- [Introduction to the AWT Framework Page](#)
- [Event Handling Techniques Page](#)
- [Packaging your Java Desktop App to a JAR File](#)

#### Unit 10 Learning Objectives:

- Students learn the principles behind GUI applications.
- Students learn how to use different containers, components and the layout manager to create GUI applications.
- Students learn how to run event handling techniques that provide functionality to the user interface components in their applications.
- Students learn how to make an executable Java file to distribute their application.