



Certificate Course In Java

Prepared by DOEACC Centre, Calicut

Syllabus & Curriculum
for Certificate Course in
Java

1. General Description

The purpose of this course is to provide a straight forward way for an absolute beginner helping him/her get their mind around java and object-oriented programming. It provides participants with the information and experience necessary to use Java to develop cross-paltform applications, and to integrate Java into web pages.

2. General Eligibility

This course is meant for a beginner looking to be a computer programmer. The participant should have general familiarity with their operating system environment and also the ability to create and edit text files and manipulate directories.

While this course does not assume that participants have any programming language experience, some prior exposure to other structural or object-oriented language is beneficial.

3. Hardware & Software required for Teaching

Software

- Java2 Platform Standard Edition (SDK 1.2 or above) installed and the Java2 Platform Standard Edition documentation. Currently Sun makes a Java Development Kit (JDK) freely available for Windows or Linux and can be downloaded from <http://java.sun.com>.
- Any text editor

Hardware

- A system with a 400MHz or faster processor with not less than 150 MB disk space and 128MB RAM is recommended. However, the documentation along with the SDK of the version being installed may be referred for up to date information.

4. Course objectives and content

Upon completion of this course participant will be able to

- ◇ Become comfortable with object oriented programming
- ◇ Develop cross-paltform applications, and to integrate Java into web pages
- ◇ Develop event-driven GUI based application
- ◇ Design and develop object-oriented solutions to simple and moderately complex problems.

5. Suggested reference material

Books

JAVA2 – The Complete Reference, Patrick Naughton, Herbert Schildt, Tata McGraw Hill Publications
The JAVA handbook, Patrick Naughton, Tata McGraw Hill Publications
JAVA-How to program, Dietel & Dietel, Prentice Hall Publication
Programming with Java: A Primer Balaguruswamy E, Tata McGraw Hill

Online Documentation

Java2 Platform Standard Edition documentation

6. Examination /Evaluation scheme

The evaluation will be done based on one theory Examination of 2 hours and practical examination of 3 hours duration.

- a. Theory Examination: The examination will be of 2 hours duration and will contain 100 objective type questions with maximum marks of 100.
- b. Practical Examination: One Practical examination of 3 hours duration will be conducted on the modules described in the curriculum. The maximum marks will be 100.

7. Suggested duration for theory, tutorials, case study & practical sessions

Module	Topic	Theory (hrs)	Lab (hrs)
1	Introduction to Java	2	4
2	Holding Data	4	8
3	Controlling the flow	6	8
7	Object Oriented Programming Concepts	10	10
8	Inheritance & Packaging	6	6
4	Handling Error/Exceptions	4	4
5	Handling Strings	4	4
9	Threads	4	4
10	I/O and Streams	4	4
11	Understanding core packages	4	4
12	Holding Collection of data	4	4
13	Java Applications	6	6
14	Introduction to Java Applets	4	4
15	Basic Networking Concepts	4	4
<i>Total Duration</i>		66	74
Project		35 hours	
Total		175 hours	

Detailed Syllabus

Contents

1. **Introduction to Java**
2. **Holding Data**
3. **Controlling the flow**
4. **Object Oriented Programming Concepts**
5. **Inheritance & Packaging**
6. **Handling Error/Exceptions**
7. **Handling Strings**
8. **Threads**
9. **I/O and Streams**
10. **Understanding core packages**
11. **Holding Collection of data**
12. **Java Applications**
13. **Introduction to Java Applets**
14. **Basic Networking Concepts**

1) Introduction to Java

- What is Java?
- Background/History of Java
- The Internet and Java's place in it
- Applications and Applets
- Java Virtual Machine
- Byte code - not an executable code
- Procedure-Oriented vs. Object-Oriented Programming
- Basics of OOP
 - Abstraction
 - Inheritance
 - Encapsulation
 - Classes, subclasses and super classes
 - Polymorphism and Overloading
- Compiling and running a simple "Hello World" program
 - Setting Up Your Computer
 - Writing a Program
 - Compiling, Interpreting and Running the program
 - Common errors

2) Holding Data

- Primitive Data Types
 - Integers
 - Floating-Point types
 - Characters
 - Booleans
- User-Defined Data Types
- Declarations
- Constants
- Identifiers, Literals
- Type Conversion and Casting
- Objects and Wrapper Classes
- Variables
 - Variable Definition and Assignment
 - Default Variable Initializations
- Command-Line Arguments
- Arrays of Primitive Data Types
- Comment Syntax
- Garbage Collection

3) Controlling the flow

- Expressions
- Using Operators

- Arithmetic, Bitwise, Relational, Logical, Assignment, Conditional, Shift, Ternary
- Auto-increment and Auto-decrement
- Using control statements
 - Selection statements
 - If, Switch
 - Loops and loop options
 - While, do-while, for
 - Jump statements
 - Break, continue and return

4) **Object Oriented Programming Concepts**

- Abstraction
- Encapsulation
- Polymorphism and Overloading
- Fundamentals of Classes
 - A simple class
 - Creating Class Instances
 - Adding methods to a class
 - Calling Functions/Methods
- Using 'this' keyword
- Constructors
 - Default constructors
 - Parameterized constructors
- More on methods
 - Passing by Value, by Reference
 - Access Control
 - Methods that Return Values
 - Method Overloading
 - Recursion
- Nested and Inner classes

5) **Inheritance & Packaging**

- Inheritance
 - Using 'extends' keyword
 - Subclasses and Superclasses
 - 'super' keyword usage
 - Overriding Methods
 - Dynamic Method Dispatch
- The Object class
- abstract and final Classes
- Packages
 - Defining a package
 - Importing a package

- Access Control
 - Interfaces
 - Defining an interface
 - Implementing and applying interfaces
- 6) Handling Error/Exceptions**
- Basic Exceptions
 - Proper use of exceptions
 - User defined Exceptions
 - Catching Exception
 - try
 - catch
 - Throwing and re-throwing
 - throw
 - throws
 - Cleaning up using the finally clause
- 7) Handling Strings**
- Creation, Concatenation and conversion of a string
 - Changing case
 - Character Extraction
 - String Comparison
 - Searching strings
 - Modifying strings
 - String Buffer
- 8) Threads**
- Create/instantiate/start new threads
 - i. Extending java.lang.Thread
 - ii. Implementing java.lang.Runnable Interface
 - Understand thread execution
 - Thread Priorities
 - Synchronization
 - Inter-thread communication, Deadlock
- 10) I/O and Streams**
- java.io package
 - Files and directories
 - Streams
 - Byte Streams and Character Streams
 - Reading/Writing Console Input/Output
 - Reading and Writing files
 - The Serialization Interface
- 10) Understanding core packages**

- Using java.lang Package
 - java.lang.Math
 - Wrapper classes and associated methods
 - Number
 - Double, Float
 - Integer, Byte
 - Short, Long
 - Character
 - Boolean
- Using java.util package
 - Core classes
 - Vector
 - Stack
 - Dictionary
 - Hashtable
 - Enumerations
 - Random Number Generation

12) Holding Collection of data

- Arrays and collection classes/interfaces
- Map/List/Set implementations
 - Map interface, List interface, Set interface
- Collection classes
 - ArrayList, LinkedList, HashSet and TreeSet
- Accessing collections/use of an Iterator
- Comparator

13) Java Applications

- Intro to AWT and Swing
- Working with Frame, windows, graphics, color, fonts
- AWT controls
 - Buttons, Checkbox, Choice, List and TextField
- Layout Managers
 - Flow Layout, Grid Layout and Border Layout
- JFrame and JPanel containers
- User Interface Events
 - Event Classes and Event Listener Interfaces
- Adapter Classes

14) Introduction to Java Applets

- What is a Java applet?
- Applet lifecycle methods
- Build a simple applet
- Using AppletViewer
- Adding Controls

➤ Animation Concepts

15) Basic Networking Concepts

- What is a Client/Server Application?
- Manipulating URL's
- Establishing a simple server
- Establishing a simple client
- Client/Server Interaction using TCP
- Connectionless Client/Server Interaction using Datagram(UDP)