



<b>JAVA: LEVEL 1.....</b>	<b>2</b>
OBJECTIVES .....	2
TARGET GROUP.....	2
TRAINING METHOD.....	3
COURSE DURATION.....	3
COURSE BREAKDOWN .....	3
<b>JAVA : LEVEL2.....</b>	<b>6</b>
OBJECTIVES .....	6
TARGET GROUP.....	6
TRAINING METHOD.....	7
COURSE DURATION.....	7
<b>JAVA : LEVEL 3.....</b>	<b>10</b>
OBJECTIVES .....	10
TARGET GROUP.....	10
TRAINING METHOD.....	10
COURSE DURATION.....	11
COURSE BREAKDOWN .....	11



## JAVA: LEVEL 1

### OBJECTIVES

The DWIT Training - Java course is targeted for beginners who want to:

- Learn how to think and write meaningful piece of code in Java.
- Learn how to read JAVAcode that has been written by somebody else.
- Learn how to map literary description of a problem (requirement) to an application/library coded in Java. In summary, this course teaches how to program using Java programming language.

This is a core basic level course that is essential for anyone who have no prior programming experience but wish to be a professional Java engineer in future

### TARGET GROUP

Anyone who has some basic knowledge about programming and wants to learn to write applications in JAVAfor any purpose e.g. curiosity, hobby, to complete an academic project, to work towards a career as JAVAprogrammer, to help in project management, etc.

#### Prerequisites

- Basic knowledge about programming, bits/bytes, procedures, classes, computer architecture, etc. If you just have a theoretical knowledge that is perfectly okay but you should have strong convictions on what programming is, and what you hope to achieve from this class.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Java (self-study and practice).
- There are no prior educational level requirement for this course. Anyone from 10+2 student to someone who is doing her PHD in Genetic Engineering is welcome to take this course.
- If you are only interested in theory and have no interest/patience in spending at least 10 hours every week throughout the duration of the course, then this course is clearly not for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six -odd months, then this class may not be for you!



## TRAINING METHOD

The course is spread over 40 hours that consists of lecture and lab work. There will be approximately 10 hours of lectures and 30 hours of hands-on lab work.

- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but specially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Java.

## COURSE DURATION

- 40 hours
- Classes  
-Morning/Evening

## COURSE BREAKDOWN

- **Theory**
- **OVERVIEW OF JAVA LANGUAGE**
  - Introduction
  - H/w and s/w requirements
  - Installation of jdk



- **PROGRAMMING WITH JAVA**
  - Class declaration
  - Members of classes
  - Structure of java class
  - Main method
  - Command line arguments
  - Source code compilation
  - Coding convention
  - Java packages
  
- **CONSTANT, VARIABLES AND DATA TYPES**
  - Primitives and non-primitives variables
  
- **DECISION AND BRANCHING**
  - IF, ELSE, SWITCH, BREAK, CONTINUE
  
- **LOOPING**
  - FOR, WHILE, DO-WHILE
  
- **FUNDAMENTALS OF LOOPS**
  - Initializing objects
  - Static members
  - Inheritance
  - Polymorphism
  - Encapsulation
  
- **ABSTRACT CLASS AND INTERFACES**
  - Defining interfaces
  - Separating interface and implementation
  - Implementing and extending interfaces
  - Abstract classes
  
- **EXCEPTION HANDLING**
  - Exceptions and the exception hierarchy
  - Throwing exceptions
  - Catching exceptions
  - Chaining exceptions
  - The finally block
  
- **ADVANCE DATA STRUCTURES (JAVA COLLECTION CLASSES)**
  - ⊗ Arrays
  - ⊗ List<e> interface and its implementation
  - ⊗ Map<k,v> interface and implementation
  - Set<e> interface and implementation



- **JDBC CONNECTION**

- ⊗ Jdbc overview
- ⊗ Using DriverManager, Connection, Statement, PreparedStatement and ResultSet
- Create, delete, insert, update statements

- **JAVA DOC AND JAVA LIBRARIES**

- ⊗ Preparing java doc
- ⊗ Exporting java doc
- ⊗ Implementing java libraries
- ⊗ String class
- Math class

### Labs

- Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem solving techniques to the students.

### DISCLAIMER

Please note that Deerwalk Institute of Technology reserves the right to change the course syllabus of DWIT Training - Java – Level 1 course at any time without prior notification.



## JAVA : LEVEL2

### OBJECTIVES

The DWIT Training - Java – Level 2 course is targeted for trainees:

- Who have had some prior beginner level hands-on programming experience in Java programming language.
- Who have programming experience in some other programming language (e.g. Java, Obj-C, PHP, C, C++, etc.) and want to learn Java .

### TARGET GROUP

- High school and university students (undergraduate, graduate, etc.) who want to do coursework (e.g. project, etc.) in JAVA.
- Someone who has experience in some other programming language (e.g. C/C++, PHP, Perl, etc.), but has never done programming in ANDROID.
- Someone who is already working as a professional VB.NET developer and wants to switch to ANDROID.
- Someone who did her undergraduate in Economics, has been working in Media sector since graduation, and also working as a professional freelance PHP developer.
- Electrical/Electronic undergraduates in their 3rd semester who want to beef up their software skills prior to graduation.

### Prerequisites

- Successfully complete the entrance test with score of at least 40% (for trainees directly applying to this level).
- Successfully complete the DWIT Training - Java – Level 1 course (not applicable to trainees directly applying to this level).
- Successfully complete the interview.
- o Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Java (self-study and practice).



## TRAINING METHOD

The course is spread over 40 hours that consists of approximately 15 hours of lecture and 25 hours of hands-on lab work.

- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.
- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

In summary, the only effective way to learn programming is to write lots of code. So in order to really make this training productive, students are encouraged to spend as much time as necessary to complete the lab exercises on time. As part of the course, students will spend at least 30 hours in the lab but specially if you are new to programming or are coming from a non-computer-science background, it is recommended that you spend at least 10-20 hours per week outside of the class on your own to practice coding in Java.

## COURSE DURATION

- 40 hours
- Classes  
- Morning/Evening



- **WEB APPLICATION BASICS**
  - How the web works
  - Http overview, brief html review
  - Overview of java ee, servlets & web applications
  
- **SERVLET API**
  - ⊗ Html forms
  - ⊗ Http: request-response, headers, get, post
  - ⊗ Overview: how servlets work
  - ⊗ Servlet lifecycle: init(), service(), destroy()
  - ⊗ Requests and responses
  - ⊗ Http servlets: httpServletRequest, HttpServletResponse and HttpServlet
  - ⊗ Deployment descriptor
  - Accessing parameters
  
- **ADDITIONAL SERVLET CAPABILITIES**
  - RequestDispatcher: including and forwarding
  - Sharing data with the request object attributes
  - Sharing data with ServletContext and ApplicationContext
  
- **JAVASERVER PAGES**
  - Basics and overview
  - Lifecycle of a jsp
  - Scriptlet , page directive
  - Model view controller (mvc)
  - Data sharing among servlets & jsp
  - Request, application, session and page scope
  - Predefined jsp implicit objects (request, session, application, page)
  - <jsp:include>, <jsp:forward>
  
- **USING CUSTOM TAGS**
  - ⊗ Custom tags to reduce jsp complexity
  - ⊗ The jstl
  - ⊗ Jsp expression language (el)
  - ⊗ Using custom tags
  - The c:url, c:param, c:foreach, c:out tags
  
- **MORE JSP CAPABILITIES AND SESSION MANAGEMENT**
  - ⊗ Http as a stateless protocol
  - ⊗ Hidden form fields
  - ⊗ Cookies: overview, api, using cookies
  - ⊗ Session overview: cookies and session tracking
  - HttpSession





- Putting data into a session object
- Retrieving data from a session object
- Using session data in servlets and jsps
  
- **ADDITIONAL TOPICS**
  - Servlet filter overview
  - Filtering examples, lifecycle, & filter chains
  - Filter api, modifying a request, modifying a response

### **Labs**

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem solving techniques to the students.

### **DISCLAIMER**

Please note that Deerwalk Institute of Technology reserves the right to change the course syllabus of DWIT Training - Java – Level 2 course at any time without prior notification.



## JAVA : LEVEL 3

### OBJECTIVES

This course builds on the foundation laid by DWIT Training - Java – Level 3 to prepare trainees for a career as Java software engineer.

### TARGET GROUP

#### Prerequisites

- Successfully completed the DWIT Training - Java – Level 3 or obtained at least 40% score on the entrance exam.
- The latter case applies for new students that are directly attempting this training.
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Java (self-study and practice).
- Please note that this is a lab intensive course where the students will be expected to work on lab exercises for approximately half the duration of the session.

### TRAINING METHOD

- The course is spread over 40 hours that consists of approximately 20 hours of lecture and 20 hours of lab work.
- Lab exercises are mandatory, have a fixed deadline, and are graded. The course puts heavy emphasis on lab exercises because software programming can only be learnt well by explicitly putting into practice the principles that have been taught (i.e. in simpler terms – by doing lots and lots of coding). Late submission (past the deadline) of exercises incur some penalty from total points.
- Instructors may provide relevant lecture/lab notes to students as (and when) necessary in the form of printed handouts and or via emails.
- Instructors may provide supplementary code snippets to students via email or in lab class to support the theory and or lab material that is being taught.



- At the end of the course, students may have to give an exam (which will be optional), that will test their knowledge on the material covered during the course. This exam may be practical and/or theoretical and is mandatory for any student wishing to join a higher level.
- Students are graded on the basis of attendance, lab exercises and exam in the increasing order of importance.

## COURSE DURATION

- 40 hours
- Classes
  - Morning/Evening

## COURSE BREAKDOWN

- **GROOVY FUNDAMENTALS**
  - Differences between groovy and java
  - ⊗ Closures
  - ⊗ Lists and maps
  - ⊗ Ranges
  - Lists, maps, and sets
- **GRAILS PROJECT STRUCTURE**
  - Overview of a grails project
  - ⊗ Directories for model classes, controllers, and views
  - Data source configuration
  - ⊗ Command line utilities
  - Ide integration
- **DOMAIN CLASSES**
  - Object-relational mapping with gorm
  - ⊗ Constraints and validation
  - ⊗ Modeling relationships
  - ⊗ Performing crud operations
  - Dynamic queries
- **SCAFFOLDING**
  - ⊗ Generating controllers
  - Generating views



- **CONTROLLERS**
  - ⊗ Setting the default action
  - ⊗ Accessing request attributes
  - ⊗ Flash scope
  - ⊗ Rendering a response
  - Redirects and returns
  
- **GROOVY SERVER PAGES**
  - Built-in grails tags
  - ⊗ Layouts and templates
  - Creating custom tags
  
- **ADDING AJAX SUPPORT**
  - Basics of ajax
  - Ajax-supporting tags
  - Rendering json and xml responses
  
- **SERVICES**
  - Transactions
  - Services and dependency injection

#### Labs

- Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem solving techniques to the students.

#### DISCLAIMER

Please note that Deerwalk Institute of Technology reserves the right to change the course syllabus of DWIT Training - Java – Level 2 course at any time without prior notification.