



Gwani Software



TRAINING DEPARTMENT
(Knowledge & Expertise)

Java Curriculum

December 2011

© Gwani Software Ltd, 2011.

All rights reserved.

Disclaimer

- Java is a trademark of Sun Microsystems Inc.,
- And any other trademark used belongs to its rightful owner.

Warning

This document is an exclusive property of Gwani Software Ltd, permission is granted to print and copy this document by trainees, instructors, supervisors and affiliated academies of Gwani Software likewise for non-commercial distribution by anyone.

With the exception of the above permission, no part of this document may be printed, copied, modified or used by anyone without a prior written permission is obtained from Gwani Software. Contravening this provision may lead to legal proceeding in a court of law.

Proposed by

Abubakar Muhammad

faqeer@gwanissoftware.com

Approved for usage by

Al-Ameen Abubakar, Director Training, this 5th day of Muharram, 1433
equivalent to

1/12/2011.

Gwani Software

TRAINING DEPARTMENT

Java

General Description: - This course is intended to give the trainee basic skills in writing Java application using JSDK tool on CLI mode.

Aims: - The aims of this course are:

1. To introduce the trainee into GUI programming in Java using lightweight and heavy weight.
2. To drill trainee on how to write programs in Java using CLI AND GUI interfaces.
3. To guide the trainee on how to design software using Java.

Objectives: - The trainee at the end of the training session should:

- Be able to write application in Java.
- Know how to use the CLI and GUI I/O of Java.

Target Audience: - This course should be taken by computer instructors, programmers, software engineers, web designers, network engineers, computer scientist and anyone interested in programming.

Pre-requisite:- Data structure, programming essentials, flowcharting and algorithm are standing pre-requisite to this course. While prior experience in C, C++ or PHP is an added advantage.

Approximate Duration: - The course requires 21 hours of class sessions with practical application design.

Method of assessment: - Trainee is to be assessed with a software design project using Java on a given case study.

Methodology: - The class takes a lesson discussed it, design and application base on the lesson, then progresses to the next lesson until all lessons are covered. The trainee will periodically be given assignments on design of Java application.

Recommendations resource materials: - The following materials are recommended for the trainee study:

1. Khawar Zaman Ahmed, CaryE. Umrysh, (2001), '**Developing enterprise Java application with J2EE, and UML,**' Addison-Wesley.
2. Asbury, Stephen & Scott Weirer, (1999), '**Developing Java Enterprise Applications**', John Wiley & Sons, New York.
3. Monson-Hacfel, Richard (1999), '**Enterprise Java Beans**', O'reilly & Assoc., California'.
4. Barry Burd, (2005), '**Beginning Programming with Java[™] for Dummies (R), 2nd Edition**', Wiley Publishing Inc., Hoboken.
5. Donald Bdes, (2002), '**Java Programming with oracle JDBC**', O'reilly.
6. Vivek Chopra, Sinj Li, Rupert Jones, Jon Eaves, John T. Bell, (2005), '**Beginning Java server pages**', Wiley Publishing Inc., Indianapolis.
7. Ibm.com/developers works, '**Introduction to Java I/O**'.
8. Mauro Mannilli, '**Professional Java User Interfaces**', John Wiley & Sons Ltd.

9. Mark Richards & IMfoq, (2006) '**Java transaction Design strategies**', c4media publisher.
10. **Java cook book**
11. Edward Morgan Taster, '**Java how to program, 4th edition**', Prentice hall
12. Ivor Horton (2005), '**Beginning Java 2nd edition**', Wiley publishing Inc., Indianapolis.
13. Donald Doherty & Michelle M. Manny, (1998), '**Teach yourself JBuilder 2 in 21 days**', Sams publishing, Indianapolis.
14. Laira Lemay, Charles L. Perkins, '**Teach Yourself Java in 21 days**', Sams publishing, Indianapolis'.
15. Doug Tidwell, (1999), '**XML Programming in Java**', Cyber evangelist.

Day	Lesson
1	Getting Started with Java: - Programming language, interpreters, compilers, JVM, historical development about Java initial intention of developing Java, how oak change to Java OOP in brief.
2	Using the Java development kits: - installing a JSDK, using MS-DOS editor, using compile command (Java) classes and objects.
3	Standard input and outputs: Outputs, system, out command print, println, printing a literal, printing argument, printing expression. Inputs, use of bufferedReader, inputStream and Inputstream (system.out). echo printing, converting inputs, to integer converting inputs to doubles, converting inputs to floats.
4	GUI inputs and outputs: Heavy weight and light weight graphics package, awt and swing, Joptionpane showInputDialog,

	showMessageDialog, date type casting, integer.parse, double.parse, float.parse, system.exit(o), practical example.
5	Passing argument: The sting [] Argument, syntax, testing for not null arguments, casting a pass argument, printing an argument, printing a casted argument, running program with an argument, using batch file to supply arguments to an application. <i>Worked example.</i>
6	GUI: Using JFrame, setsize, setTitle, setVisible, setLocation, using JFrame and JOption. <i>Worked Example.</i>
7	Date and Simple Date: java.util. new date object, SimpleDateFormat, toString, display date, symbol: y, m, D, h, H, a, m, s, S, E
8	Numerical Data: Variable, expression, data types byte, short, int, long, float, double, arithmetic operators, procedure type casting, numeric promotion, constant, use of final statement parseInt, parseLong, parseFloat, ParseDouble, operator overloading, the Maths class, methods for commonly used mathematical functions, Decimal Format.
9	Calendar class: The java.util.Date class, GregorianCalendar class, calendar.month, Constants: YEAR, MONT, DATE, DAY_OF_MONTH, DAY_OF_MONTH, DAY_OF_WEEK, WEEK_OF_YEAR, WEEK_OF_MONTH, AM_PM, HOUR_OF_DAY, MINUTE, Calendar.getTime(), Calendar.get(constants).
10	Defining a User Class: Application with only one class and are method i.e. the main, reason for using more than one class for large application; instantiable class, Data member, instance variable, instance data value, visibility modifier, class constructors, information hiding, client programs, encapsulation, local variables, returns variable, parameter passing, Accessors, mutators and overloaded methods.

11	<p>Selection Statements: Sequential execution, control statement, if statement, relational operators, block if statement, if statement, if statement with else, nested if statement, Boolean expressions, Boolean operators, Short circuit evaluation comparing objects.</p> <p>Switch statement, syntax, default, break, and continue.</p>
12	<p>Loops: Definition, importance, While, do-While, for statement, recursion, endless loop, priming read pretest, shorthand assignment operators, confirmation, dialog box, control variable, nested for statement.</p>
13	<p>Drawing Graphics: Review of JFrame, container, awt and swing classes, setVisible, setSize, setLocation, container, setBackground, placing buttons, event handling, JLabel, JTextField, JTextArea, Menus.</p>
14	<p>Characters and String: Char, ASCII, (int), (char) Strings.shorthand, toupperCase, out-of-boundexpression, compareTo, replaceAll, StringBuffer, CharAt, the matches method (equal and matches).</p>
15	<p>Arrays: Basics, declaration, initialization, single dimensional, length, fixed-size array declaration, variable-size array declaration, insertion into array deletion from array.</p>
16	<p>File Processing: File output and input file, current directory, JFileChooser, file filter, low-level file i/o, stream, source destination, FileOutputStream, data catching, data buffer. FileInputStream.</p>
17	<p>File Processing Continues: High-level file i/o, DataOutputStream, DataInputStream, binaryfile, textfile, printWriter object i/o.</p>
18	<p>Introduction to OPP: Inheritance and polymorphism, polymorphism defining classes with inheritance, sibling classes benefits of polymorphism, inheritance and constructors.</p>
19	<p>Introduction to OPP: Inheritance, inheritance vs. interface challenge for</p>

	student, 'computing course grades' to complete the lesson.
20	Exceptions: Catching exception, try-catch, throwing exception, propagation exceptions, exceptions thrower, exception catcher, exception propagator, types of exception (Runtime exception and IOException).
21	Revision.

GWANI Software