

ITT Technical Institute
IT218
Programming in Java I
Onsite Course

SYLLABUS

Credit hours: 4

Contact/Instructional hours: 50 (30 Theory Hours, 20 Lab Hours)

Prerequisite(s) and/or Corequisite(s):

Prerequisite: IT104 Introduction to Computer Programming or equivalent

Course Description:

This course covers the fundamentals of Java programming. Object-oriented programming techniques and Unified Modeling Language (UML) are also introduced. Students practice how to build Java classes, graphical user interfaces, and event driven programs. They also explore how to write Java codes that use arrays, strings, file input and output, and exception handling.

Syllabus: Programming in Java I

Instructor:	_____
Office hours:	_____
Class hours:	_____

Major Instructional Areas

1. Introduction to computers, programming, and Java
2. Primitive data types and operations
3. Selection statements
4. Loops
5. Methods
6. Arrays
7. Objects and classes
8. Strings and text I/O
9. Inheritance and polymorphism
10. Abstract classes and interfaces
11. Object-oriented design
12. GUI Basics
13. Graphics
14. Event-driven programming
15. Creating user interfaces
16. Applets and multimedia

Course Objectives

1. Describe the fundamentals of the Java programming language.
2. Write Java programs by using primitive data types and operations.
3. Write Java programs by using selection statements.
4. Write Java programs by using loop structures.
5. Write Java programs by using user-defined methods.
6. Write Java programs by using arrays.
7. Write Java programs by using user-defined objects and classes.
8. Write Java programs by using strings and text I/O.
9. Implement inheritance and polymorphism in Java programs.
10. Write Java programs by using abstract classes and interfaces.
11. Develop applications using the object-oriented design approach.
12. Develop GUIs (Graphical User Interfaces) for Java applications and applets.
13. Develop graphics components for Java applications and applets.
14. Implement event-driven programming in Java programs.
15. Create user interfaces using the various Swing components.
16. Create Web applications by using Java applets.

SCANS Objectives

SCANS is an acronym for Secretary's Commission on Achieving Necessary Skills. The committee, created by the National Secretary of Labor in the early 1990s, created a list of skills and competencies that the committee feels are necessary for employees to function in a high-tech job market.

1. Acquire information.
2. Know how technological systems work and function effectively.
3. Demonstrate competence in understanding systems.
4. Know how the structures of a system relate to goals.
5. Demonstrate competence in selecting technology, including determining desired outcomes and applicable constraints.
6. Demonstrate competence in applying technology to task.
7. Design and implement applications to solve business problems.
8. Identify appropriate technology to solve business problems.

Course Outline

Note: All graded activities, except the Project, are listed below in the pattern of <Unit Number>.<Assignment Number>. For example, Labs: 3.2 refers to the 2nd lab activity in Unit 3.

Unit	Activities
1— Introduction to Java Programming	<ul style="list-style-type: none"> • Content Covered: <ul style="list-style-type: none"> <i>Introduction to Java Programming:</i> <ul style="list-style-type: none"> ○ Chapter 1, “Introduction to Computers, Programming, and Java” ○ Chapter 2, “Elementary Programming” • Labs: 1.1, 1.2 • Homework Exercises: 1.1, 1.2
2— Control Program Execution	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming:</i> <ul style="list-style-type: none"> ○ Chapter 3, “Selections” ○ Chapter 4, “Loops” • Labs: 2.1, 2.2 • Homework Exercises: 2.1, 2.2
3— User-Defined Methods and Arrays	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming:</i> <ul style="list-style-type: none"> ○ Chapter 5, “Methods” ○ Chapter 6, “Arrays” • Labs: 3.1, 3.2 • Homework Exercises: 3.1, 3.2
4— Classes and Objects	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming:</i> <ul style="list-style-type: none"> ○ Chapter 7, “Objects and Classes” • Labs: 4.1, 4.2 • Homework Exercises: 4.1 • Projects: Project 1 (Start)
5— Strings and Text Input/Output	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming:</i> <ul style="list-style-type: none"> ○ Chapter 8, “Strings and Text I/O” • Labs: 5.1, 5.2 • Homework Exercises: 5.1, 5.2 • Projects: Project 1 (Submit)
6— Inheritance and Polymorphism	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming:</i> <ul style="list-style-type: none"> ○ Chapter 9, “Thinking in Objects” ○ Chapter 10, “Inheritance and Polymorphism” • Labs: 6.1, 6.2 • Homework Exercises: 6.1 • Projects: Project 2 (Start)
7— Abstract Classes and Interfaces	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming:</i> <ul style="list-style-type: none"> ○ Chapter 11, “Abstract Classes and Interfaces” ○ Chapter 12, “Object Oriented Design and Patterns” • Labs: 7.1, 7.2

Unit	Activities
	<ul style="list-style-type: none"> • Homework Exercises: 7.1 • Projects: Project 2 (Submit)
8— GUI and Graphics Programming	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming</i>: <ul style="list-style-type: none"> ○ Chapter 13, “GUI Basics” ○ Chapter 14, “Graphics” • Labs: 8.1, 8.2 • Homework Exercises: 8.1 • Projects: Project 3 (Start)
9— More GUI and Event-Driven Programming	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming</i>: <ul style="list-style-type: none"> ○ Chapter 15, “Event-Driven Programming” ○ Chapter 16, “Creating User Interfaces” • Labs: 9.1, 9.2 • Homework Exercises: 9.1 • Projects: Project 3 (Submit)
10— Applets	<ul style="list-style-type: none"> • Read from <i>Introduction to JAVA Programming</i>: <ul style="list-style-type: none"> ○ Chapter 17, “Applets and Multimedia” • Labs: 10.1, 10.2 • Homework Exercises: 10.1 • Projects: Project 4 (Start)
11— Review and Final Exam	<ul style="list-style-type: none"> • Final Exam • Projects: Project 4 (Submit)

Instructional Methods

In this course, you will write Java programs to create simple applications. This will help you examine the basic Java language concepts and the Java environment and create applets.

The following are some of the strategies used in the course:

- There are code demonstrations.
- Open-ended questions are used to initiate discussions in class.
- Labs and projects will give you significant hands-on practice.
- Each unit includes two lab exercises that reinforce the content specific to that unit. Programming requires you to apply concepts presented earlier in the course to later units.
- Each unit includes a homework assignment that requires you to submit answers to multiple-choice questions or to perform research using the ITT Tech Virtual Library.

Classroom activities will allow you to participate on a team to identify the structures and features of Java applications. You will also be expected to complete writing assignments and labs in every unit. In addition, a project will be assigned in every other unit.

Instructional Materials and References

Student Textbook Package

- Liang, Y. D. (2011). *Introduction to java programming comprehensive version* (Custom 7th ed.). Boston, MA: Pearson Custom.

- Liang, Y. D. (2011). *Introduction to java programming VideoNotes website* (Custom 7th ed.). Boston, MA: Pearson Custom.
- Liang, Y. D. (2010). *Programming in java student CD* (Custom ed.). Boston, MA: Pearson Custom.
- Liang, Y. D. (2010). *Programming in java II student CD* (Custom ed.). Boston, MA: Pearson Custom.

References

ITT Tech Virtual Library

Log on to the ITT Tech Virtual Library at <http://library.itt-tech.edu/> to access online books, journals, and other reference resources selected to support ITT Tech curricula.

Books

You may click “Books” or use the “Search” function on the home page to find the following books.

- Books 24x7
 - Horton, Ivor. *Ivor Horton’s Beginning Java 2, JDK*. 5th ed. Indianapolis, IN: Wiley Publishing, Inc., 2005.
 - Jarc, Duane. *Learning Java Through Applications: A Graphical Approach*. Hingham, MA: Cengage Charles River Media, 2005.
 - Kurniawan, Budi. *Java 5: A Beginner’s Tutorial*. Vancouver, BC, Canada: BrainySoftware.com, 2006.
 - Levenick, James. *Simply Java: An Introduction to Java Programming*. Hingham, MA: Cengage Charles River Media, 2006.
 - Schildt, Herbert. *Java: The Complete Reference, J2SE*. 5th ed. Emeryville, CA: McGraw-Hill/Osborne, 2005.
 - Sestoft, Peter. *Java Precisely*. 2nd ed. Cambridge, MA: [The MIT Press](http://TheMITPress.com), 2005.

School of Information Technology

- Recommended Links
 - Introduction to Programming Using Java
- Tutorial Links
 - Tutorialized> Java> Miscellaneous

Other References

The following resources may be found **outside** of the ITT Tech Virtual Library.

Web sites

- Java Programming Notes
<http://www.leepoint.net/notes-java/index.html>
A collection of Java lessons categorized by topic
- Java Tutorial
<http://java.sun.com/docs/books/tutorial/collections/intro/index.html>
A tutorial on the official Java Web site maintained by Sun Microsystems
- RoseIndia Java Tutorial
<http://www.roseindia.net/java/jdk6/index.shtml>
A comprehensive collection of Java tutorials from beginning to advanced levels

All links to Web references outside of the ITT Tech Virtual Library are always subject to change without prior notice.

Course Evaluation and Grading

Evaluation Criteria Table

The final grades will be based on the following categories:

CATEGORY	WEIGHT
Homework Exercises	20%
Labs	35%
Projects	25%
Final Exam	20%
Total	100%

Note: Students are responsible for abiding by the Plagiarism Policy.

Grade Conversion Table

The final grades will be calculated from the percentages earned in the course, as follows:

A	90–100%	4.0
B+	85–89%	3.5
B	80–84%	3.0
C+	75–79%	2.5
C	70–74%	2.0
D+	65–69%	1.5
D	60–64%	1.0
F	<60%	0.0