



UNIVERSITY OF NEW YORK IN TIRANA Course Syllabus

Course: Java (4 credit hours)
Professor: Narasimha Rao V
Office Hours: Monday (10 AM – 4 PM).
Phone: 069240 1656
E-mail: narasimharao@unyt.edu.al

Course Description

This module develops problem-solving and programming skills essential in professional programming using the Java language.

Catalog Description

This course deals with the basics of programming in Java. Apart from the basics, the course also deals with advanced topics such as Swing and JDBC.

Prerequisite: Programming in C++.

Course Aims

Object Oriented Programming (OOP) has become the predominant technique for writing software in the past decade. Many other important software development techniques (e.g. modern distributed systems and component technology) are based upon the fundamental ideas captured by object-oriented programming. This unit will introduce you to programming in a language called Java and will show you how to write well-structured object oriented programs.

Learning Outcomes

At the end of this module you will:

1. Understand the underlying principles and major benefits of Object-Oriented Programming (OOP).
2. Understand how object-oriented concepts are incorporated into the Java programming language.
3. Understand the differences between a well-structured and poorly structured program, and understand the benefits of a well structured program.
4. Be able to develop a program to solve simple “real-world” problems.
5. Be able to program in Java up to a reasonable standard.

Text Book and Development Program

Java: How to Program, H.M.Deitel & P.J. Deitel - 6th Edition, ISBN 0-13-101621-0.

The book includes a CD from which you can download the Java SDK on to your own machine. If you have any trouble installing the program, please let me know.

Other References

Lists of web resources, additional articles and books will be provided during the course.

Recommended Prior Knowledge

It is recommended that you have basic knowledge of object orientation and simple programming.

Software Installations

You must install the JDK prior to beginning this course. An installation ‘cheat sheet’ is provided in week 1 that will help you install the JDK directly from the Sun Microsystems website. Alternatively, you may install the JDK from the installation CD provided with your book.

Syllabus Outline

- Introduction to Java
- Object-Oriented Programming
- Java terminology
- Programming Concepts & Structures
- Introducing Data
- Arithmetic operators
- Method calls
- Type conversions
- Selection
- Loops
- Inheritance
- Encapsulation
- Polymorphism
- Exception Handling
- Arrays
- File Handling
- GUIs
- Abstract methods and classes
- Interfaces, Swing
- Applets

Lecture 1 - Introduction to Java

In this lecture we will look at the origins of Java and the concept of Object-Oriented Programming (OOP), the terminology involved (objects, classes and inheritance, fields, methods, constructors, method calls etc.), and we will look at our first Java program.

Reading list:

1.9, 1.13 - 1.15, 2.1 - 2.3, 3.1 - 3.7, 8.1 - 8.9.

Workload:

To be submitted at end of class

Lab Assignment -1 questions.

To be submitted next week

Home Work Assignment (Theoretical Questions)

Home Work Assignment (Lab based Questions)

Lecture 2 - Data, Data Types and Input

This seminar will look at the nature of *data* in Java programs and how to introduce data into Java programs (declaration, assignment and initialization), *data types*, and most importantly data input (we do output in seminar 1). We will also look briefly at *wrapper* classes.

Reading list:

3.6 - 3.8, 4.9, 14.8, 17.2.

Workload:

To be submitted at end of class

Lab Assignment -2 questions.

To be submitted next week

Home Work Assignment-2 (Theoretical Questions)

Home Work Assignment-2 (Lab based Questions)

Lecture 3 - The Math Class, Type Conversion and Strings

This lecture will start with a quick overview of the arithmetic operators introduced incidentally in seminar 2, and we will also look at the Math class, which contains some useful arithmetic. We will then revisit method calls just to make sure that we all understand the mechanism. Next we will look at some new material, starting with type conversions to allow us to do so called *mixed mode* arithmetic, and then we will spend the rest of our time having a more detailed look at strings (first introduced in seminar 1) and at the String class in particular.

Reading list:

6.1 - 6.8, 6.11 - 6.12, 6.15, 29.1 - 29.3, 17.2

Workload:

To be submitted at end of class

Lab Assignment -3 questions.

To be submitted next week

Home Work Assignment-3 (Theoretical Questions)

Home Work Assignment-3 (Lab based Questions)

Lecture 4 - Selection and Repetition

By now we now have a good grounding in the concept of OOP and all the basics required to write Java programs. However, so far our programs have been rather flat, i.e. they are all very straightforward with a single *flow of control* path passing through them. In this seminar we will see how we can introduce points into this flow of control path so that we can have several paths going through a program. We will also look at repetition and repetitive commands. However, before we can look at selection and repetition in detail we must first understand the concepts of expressions and statements, so we will be handling these concepts in this seminar and we will build on the previous 3 seminars to enhance this discussion.

Reading list:

4.1 - 4.7, 4.11 - 4.13, 4.16, 5.1 - 5.7, 5.8 - 5.9, 5.11 - 5.12

Workload:

To be submitted at end of class

Lab Assignment -4 questions.

To be submitted next week

Home Work Assignment-4 (Theoretical Questions)

Home Work Assignment-4 (Lab based Questions)

Lecture 5 - More OOP and Exception Handling

In this seminar we will look at some more advanced concepts of OOP such as inheritance, polymorphism and encapsulation. We will also introduce exception handling which will be used to make our programs more robust.

Reading list:

9.1 - 9.6, 10.1 - 10.4, 13.1 - 13.14

Workload:

To be submitted at end of class

Lab Assignment -5 questions.

To be submitted next week

Home Work Assignment-5 (Theoretical Questions)

Home Work Assignment-5 (Lab based Questions)

Lecture 6 - Arrays

With what we now know, we can write fairly sophisticated programs comprised of many methods and interacting classes. Individual methods may perform simple arithmetic tasks, define alternative paths through the flow of control, or cause a sequence of statements to be repeated many times. What holds us back now is the nature of the data items we are working with. These are all *scalar*, and what we would like to do is to be able to use more complicated data structures. In this seminar, we concentrate on a particular kind of higher-level data structure known as an array. We will also be taking a look at file handling towards the end of this seminar. This will allow our programs to store their output or to take their input from an alternative source other than the keyboard.

Reading list:

7.1 - 7.9, 7.11 - 12, 14.1 - 14.7, 8.7, 11.1 - 11.20

Workload:

To be submitted at end of class

Lab Assignment -6 questions.

To be submitted next week

Home Work Assignment-6 (Theoretical Questions)

Home Work Assignment-6 (Lab based Questions)

Lecture 7 - Swing Components and Classes

In this and the final seminar, we will be looking at two other aspects of Java which we have not previously considered - Graphical User Interfaces (GUIs) and Applets. This week we will be primarily looking at GUIs. However, to understand GUIs, we must first understand other important Java concepts - *abstract methods* and *classes*, and *interfaces*. These will be discussed first, and we will then move on to the interesting graphical components of Java Swing and Applets.

Reading list:

12.1 - 12.7, 11.1 - 11.13

Workload:

To be submitted at end of class

Lab Assignment -7 questions.

To be submitted next week

Home Work Assignment-7 (Theoretical Questions)

Home Work Assignment-7 (Lab based Questions)

Seminar 8 - Applets

This last seminar revolves around one of the most interesting features of Java, *applets* (small primarily web-based applications). Applets are comparatively easy to understand, yet we do first have to consider WWW mark-up languages (namely HTML). Strictly, this is not part of the syllabus for the module, but you cannot use an applet without embedding it into a WWW page. Therefore some slight knowledge of HTML is necessary. Thus, we will commence by presenting the minimum amount of HTML required to incorporate an applet into a WWW page. We will then go on to present our first applet, after which we will briefly mention some applet security issues.

At this point in the lecture, the basics will have been covered. If time allows, we will go on to look at a few more complicated examples illustrating how we can use OOP in relation to applets and how we can draw pictures within our applets.

Reading list:

Chapter 20

Workload:

To be submitted at end of class

Lab Assignment -8 questions.

To be submitted next week

Home Work Assignment-8 (Theoretical Questions)

Home Work Assignment-8 (Lab based Questions)

Concluding Remarks

By the end of this course, I hope you will have learned some of Java's power and flexibility features. I also hope you will understand the concepts behind the language.

Java is such a large, robust language, and I hope our brief exploration will have made you want to experiment further with the ideas and techniques that we have discussed.

As programming and programming languages continue to evolve, you will find that the concepts of OOP which you have learned here will allow you great flexibility as you continue to program, and combined with the Java language, it will provide you with opportunities for many interesting and exciting new developments in the future.

Course Requirements

Participation: Participation extends beyond mere attendance. Expect your instructor to keep track of how often you contribute to class discussion (as a whole), particularly during the panel discussion section. You may miss up to three classes without penalty - your first two absences count whether you have a good excuse or not. Each absence beyond the first three will cost you points off of your participation grade. The only exceptions to this rule are severe illness (doctor's note required) and UNYT approved trips/activities. Appropriate documentation for absences beyond the first three is necessary the class day directly before or after the one you miss. In general: this class is intensive and interactive. Missing class could seriously affect your grade! Students are reminded not to approach the instructor for copies of the previous week's materials during immediately before, during, or immediately after class. Students are expected to collect materials from their classmates or see the instructor during consultation hours.

Exams: Two examinations will be taken, a midterm and a final exam covering all course content during the final examination period. Test format may combine a mixture of short answer, true/false, matching, sort answer, and one or two essay questions covering *all* readings, lecture, hand-out and class discussion content.

Final Examination: To be Announced

General Requirements

Late assignments and absence from tests will *not* be tolerated. *In the event of illness or emergency, contact your instructor IN ADVANCE to determine whether special arrangements are possible. The University's rules on academic dishonesty (e.g. cheating, plagiarism, submitting false information) will be strictly enforced. Please familiarize yourself with the STUDENT HONOUR CODE, or ask your instructor for clarification.*

Criteria for Determination of Grade, including Evaluation Methods

Quizzes	15%
Assignments(Home & Lab Practice)	15%
Midterm	30%
Final	40%

Grading Scale

Letter Grade	Percent (%)	Generally Accepted Meaning
A	96-100	Outstanding work
A-	90-95	
B+	87-89	Good work, distinctly above average
B	83-86	
B-	80-82	
C+	77-79	Acceptable work
C	73-76	
C-	70-72	
D+	67-69	Work that is significantly below average
D	63-66	
D-	60-62	
F	0-59	Work that does not meet minimum standards for passing the course

Date: November 3, 2007.

Prepared by: Narasimha Rao V.