# Java: Discovering Its Power, ELENG X436.2

2 Semester Credits in EECS

## Course Description

This course presents Oracle/Sun Microsystems' Java programming language, an exciting tool for developing a variety of computer software. Widely known for its ability to bring active content to the World Wide Web, Java is now the leading general-purpose programming language. Introduction to Java Programming therefore teaches you not only how to use the Java language but how to use it well, with an emphasis on best practices and compile-time detection of bugs. Starting with the fundamentals of the Java language, we introduce the notions of effective data structures, algorithms, and object-oriented programming. From this foundation, we proceed to examine Java's rich set of standard libraries, including those for networking, graphics, database access, and more. Real applications are frequently discussed.

## Prerequisites

Working familiarity with at least one other programming language is required. An introductory course in computer programming is recommended but not required.

## Learning Outcomes

After successfully completing this course, you will:

* understand Java and its relationship to other Internet-related technologies;
* be able to write computer programs using the Java language;
* be able to apply the concepts of object-oriented programming while designing software;
* know how to use Java's strongly typed syntax and encapsulation rules correctly to produce programs with few errors;
* understand the fundamentals of good data structure design;
* be familiar with the wide variety of services offered by the Java standard libraries.

## Course Materials and Technical Requirements

### Required Materials

* *The Java Programming Language* by Ken Arnold and James Gosling, 4th Edition (Addison-Wesley, 2006).

### Recommended Reading

* *Code Complete by Steve McConnell (2nd Edition, Microsoft Press, 2004)*
* *Thinking in Java* by Bruce Eckel (2nd Edition, Prentice Hall, 2000)
* *Just Java 2* by Peter Van Der Linden (5th Edition, Prentice Hall, 2002)
* *UNIX Network Programming, Vol. I* by W. Richard Stevens (Prentice Hall, 2003)
* *UNIX Network Programming, Vol. II* by W. Richard Stevens (Prentice Hall, 1998)
* *Java in a Nutshell* by David Flanagan (6th Edition, O'Reilly and Associates, 2014)
* *Java: A Beginner's Guide* by Herbert Schildt (6th Edition, Oracle Press 2014)

### Required Java Materials

A Java compiler (available free from <http://java.sun.com/javase/downloads>) supporting the Java API; (see Module 1 Overview for detailed instructions)

A Java API guide (available free from [http://java.sun.com](http://java.sun.com/javase/reference/api.jsp?intcmp=3174) as hypertext pages). *Java in a Nutshell* contains a printed version of these documents, which is very handy. The hypertext format is more useful while you are coding, however.

Additional material will be available online, both provided by the instructor and available on the Web.

### Technical Requirements

This course is built on a Learning Management system (LMS) called Canvas and you will need to meet these [computer specifications to participate within this online platform.](http://guides.instructure.com/s/2204/m/4214/l/82542-what-are-the-basic-computer-specifications-for-canvas" \o "Confirm your computer meets Canvas specifications)

#### Optional

Canvas allows you to record audio or video files of yourself and upload them in the course. Although doing so is not required for any of the activities, using these features will enhance your engagement in the course. If you would like to use these features, you will need to have a webcam and a microphone installed on your computer.

## Learning Activities

This course consists of ten modules of study outlined at the end of this syllabus.  Each module includes a combination of:

* a reading assignment from the required text;
* reading my commentary, which provide explanation and examples on each topic;
* programming assignments;
* class discussions on the Discussion Assignments forum;
* study question assignments.

You will also submit a final project which you must pass in order to pass the course.

### Reading Assignments

Reading Assignments include both readings from the textbook as well as the instructor's commentary. The instructor's commentary on the assigned material is included online in each module. The commentary provides guidance through the assigned material and helps students focus on key concepts. Commentaries may also provide links to other related Internet resources.

After reading the relevant information for each lesson and considering the short answer questions, you may still have questions. Please visit the Student Lounge discussion forum to see if any other students have similar concerns. You may wish to post your questions there. Please also feel free to contact me directly using our course email.

### Programming Assignments

For Modules 2-9 you will submit a graded programming assignment. Detailed instructions on the assignment can be found on the Module's Overview page. You will submit your assignments by uploading the file(s) within the Learning Management System in each module. If you have more than four files to send, please compress it using ZIP format. I may ask you to re-send your attachment in a different format if I cannot read it.

Always keep a copy of your work in case a problem arises. Check returned evaluated assignments carefully, note all comments, and make the suggested corrections. This process will help prepare you for later assignments and the final project..

### Discussions

### The Discussion Assignments forum provides a vehicle for classroom interactivity based on questions created specifically for each module. All posts are "public," that is, available for all class participants to read and comment on.

### We'll have an assigned topic to discuss for every chapter we read. The discussion assignments are graded, so skipping any of them will affect your course grade. In order to encourage you to interact with your fellow students, I only award the highest discussion assignment scores to students who not only complete their own assignment but also respond to at least one other student's submission.

### When a module has a discussion assignment, please respond to the questions in a manner that reflects critical thinking. Please feel free, as well, to positively critique and offer leads and suggestions to comments and questions that other students have made.

### Each of us brings a unique perspective to this class based upon our life experiences and previous studies. But because of the continuous, open-enrollment nature of this course, at times it might be hard to sustain an actual conversation in the discussion threads with your fellow students. So instead, let's look to the discussion threads as a place to connect our multiple perspectives and construct an evolving knowledge base. The process will enrich your own studies, and the words you leave behind will help students who join the class even after you've gone on to new pursuits.

### Study Questions

### All modules also have a set of study questions, which you can find on the module's overview page. I have provided drop boxes in each module for you to submit your answers to me. Although these assignments will not be graded, I will review your answers to check your progress, and I will provide feedback as necessary to help you in future assignments including the final project.

### Final Project

For the final project, you are to design and implement a Java programming project of your choice, of between 300 and 1,000 lines of code. The project should illustrate best programming practices as discussed in this course. It must not have any dependencies on outside software (databases or networks, for example); it must be free-standing. The project should above all do something moderately useful or fun or interesting. Originality counts.

Your project is due toward the end of the course. However, a proposal is due with your submitted assignment for Module 6 and must receive my approval before you begin work on the project itself.

## Communication and Office Hours

You can always get in touch with me during the course. You can access course email by clicking on the Inbox link on the Corner Help toolbar (see also [Canvas Overview Video](http://guides.instructure.com/s/2204/m/4210/l/141852-canvas-overview-video)). You can expect me to respond to email within 48 hours of receiving messages unless I have notified the class otherwise (e.g., because of vacation or other reasons).

Please note: all course communication between students/instructor must occur within the course.

## Grading and Course Policies

Final grades will be assigned according to the following percentages:

* Discussion Assignments, 10%
* Programming Assignments, 50%
* Final Project, 40%

You must pass the final project with at least 70% to pass the course.

### Grading Information

Final grades follow the UC Berkeley grading system:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Letter Grade** | A | A- | B+ | B | B- | C+ | C | C- | D+ | D | D- | F |
| **Percentage** | 100-94 | 93-90 | 89-86 | 85-83 | 82-80 | 79-76 | 75-73 | 72-70 | 69-66 | 65-63 | 62-60 | < 60 |

Table 2: UC Berkeley Grading Systems

To [view your final grade and request official transcripts](http://extension.berkeley.edu/), login to your student account and go to "My Enrollment History."

When I grade your writing assignments, I'll be looking at content, organization, and mechanics. Please keep the following criteria in mind:

|  | **Poor** | **Needs Improvement** | **Meets Expectations** | **Exceptional** |
| --- | --- | --- | --- | --- |
| **Content** | Poor writing style with little or no specific details, no evidence of having studied the material, and/or off topic. | Adequately written; some points elaborated but with minimal use of concepts from the material. | Well written, most points elaborated with clear and detailed information that supports thoughts and ideas and uses concepts from the material. | Well written, fully elaborates points. Clear and detailed information supports thoughts and ideas and shows full acquisition of concepts from the material. |
| **Organization and Mechanics** | Little or no structure present. Grammatical errors interfere with comprehension. | Organization present but awkward. Some grammatical errors present. | Good organization with few statements out of place. Minor grammatical errors. | Clearly organized and remains focused. Few or no grammatical errors. |

Table 3: Criteria for Writing Assignments

In the Discussion Assignments forum, I'll also be looking for evidence of participation:

|  | **Poor** | **Needs Improvement** | **Meets Expectations** | **Exceptional** |
| --- | --- | --- | --- | --- |
| **Participation** | Minimal posts in number or length. Posts show little or no reflection on the topics or previous posts. | Posts address the topic but consist mostly of a rote repetition of the study materials. Little or no reflection on previous posts. | Posts address the topic with reflection. Many responses build on previous posts. | Posts show a genuine interest in contributing to the overall life of the forum. |

Table 4: Evidence of Participation for the Discussion Assignments Forum

### DSP Accommodations

If you are a student with special needs and haven't already contacted the [Disabled Student Services](http://extension.berkeley.edu/static/studentservices/career/#disabled) (DSS), please contact the office right away. Be sure to review our detailed DSP accommodations instructions.

## Academic Integrity, Research, and Proper Citation

As an online student, you are encouraged to reach out to your fellow students in the online classroom to discuss materials and ask each other questions, but there are limits to this collaboration. Reviewing lecture and reading materials and studying for exams can be enjoyable and enriching things to do with fellow students. This is recommended. However, unless otherwise instructed, homework assignments are to be completed independently, and materials submitted as homework should be the result of your own independent work.

As a UC Berkeley student you are bound by the [Academic Integrity, Research and Proper Citation policies](http://extension.berkeley.edu/upload/academic_integrity.pdf) outlined in the [UC Berkeley Extension Code of Student Conduct Policy Statement](http://extension.berkeley.edu/upload/studentconduct.pdf) dated July 11, 2013 that clearly defines what constitutes cheating, as well as plagiarism and other forms of academic misconduct.

You must review all sections of the Academic Integrity Pledge and Course Policies Module within your Canvas course and complete the following item prior to gaining access to course content:

Take the Pledge to Academic Integrity

## Course Evaluation and Course End Date

### Course Evaluation

UC Berkeley Extension is committed to improving our online courses and instruction. Please take a few minutes to participate in our course evaluation in Module Nine, to share your opinions about this course. We are interested in your online learning experience, and your candid feedback will help us plan for the future and make improvements. Please complete the evaluation before your course End Date. The evaluation does not request any personal information, and your responses will remain strictly confidential.

### Course End Date

Your access to the online classroom will expire on the course End Date, which is indicated in the initial e-mail you received when you enrolled.

With this course, you have the freedom to design your study time to meet your schedule. However, this style of learning also requires dedication and commitment to ensure that you get the homework and assignments done in a timely fashion. Use these tips to stay on track and get the most from this class.

* Get started as soon as possible: Students who submit their first assignment within the first month are more likely to complete the course than those who delay. Please do not wait to submit everything toward the end of the class. Assume that I will need a week to grade each assignment and provide feedback that will help you complete subsequent assignments.
* Create a planning calendar: Plan your homework and submittal dates, and stick with them. Students who submit assignments regularly are more likely to complete the course than those who do not. Take the time now at the beginning of the course to plan your study time by using the course Calendar tool.
* Ask questions: You are always encouraged to ask questions. Use course mail to ask me questions, ask for feedback, or just to request encouragement. I'm always pleased to be of assistance. Use the Student Lounge forum for general questions about the course or assignments that other students might need answered as well.
* Submit complete assignments: Submit only completed assignments. If you are unsure about a question, e-mail me and ask me first before submitting the assignment. I'm always pleased to be of assistance. Any incomplete assignments will be returned to you to complete and resubmit.

## Canvas Tech Support and UC Berkeley Extension Student Services

### Canvas Tech Support

The learning management system (LMS) used in this course is Canvas, which has convenient mobile apps for phones and tablets. Part of the orientation materials in your course will help you make sure that your computer is at par with Canvas specifications. Anytime you are in Canvas you can report problems, get support, and search Canvas user guides from the Help link on the top menu bar. Other options:

* Canvas Support 24/7 Hotline:  855-308-2758
* Email: [msupport@instructure.com](mailto:msupport@instructure.com)

### UC Berkeley Extension FAQs and Student Services

Start at the [Student Services webpage](http://extension.berkeley.edu/static/studentservices/) to find help with issues such as the following:

* Course registration
* Exam proctoring services
* Refunds, withdrawals, and transfers
* Grade options
* Requests for transcripts or official receipts

If you need further help, UC Berkeley’s [Extension Contact page](http://extension.berkeley.edu/static/online/#contact) lists Program Coordinator’s email and phone numbers for each academic department.

## Course Outline

You'll find complete instructions for your assignments within the course modules.

**Module 1: Introduction to the Java Programming Language**

Reading

* Gosling and Arnold, Chapter 1 introduction, (sections 1.1-1.6)
* Introducing the Java Platform
* The Java Technology Platform

Commentary

1.1 Prologue

1.2: Enter Oak

1.3: The World Wide Web

1.4: Introducing the Java Programming Language

1.5: The Function "main"

1.6: Method Calls

1.7: A Word about Code Format

**Module 2: Basic Programming Using Java**

Reading

* Arnold, Gosling, and Holmes, Chapters 7, 9, 10, and 13 (sections 13.1, 13.2, and 13.4 only)

Commentary

2.1: Prologue

2.2: The Basics of Java Syntax

2.3: Functions

2.4: Comments

2.5: Control Structures

2.6: Arrays

2.7: Strings

2.8: Basic I/O in Java

2.9: Objects

**Module 3: Software Organization and Design Interfaces**

Reading

* Arnold, Gosling, and Holmes, Chapter 6 on enumeration types [optional]

Commentary

3.1: Prologue

3.2: Functions

3.3: Locality of Reference

3.4: Recursion

3.5. Constant Values

3.6: Enumerations

3.7: Type Safety Revisited

**Module 4: Introduction to Objects and Data Structures**

Reading

* Java in a Nutshell, Chapter 3
* Gosling and Arnol, Chapters 2 and 18

Commentary

4.1: Prologue

4.2: Constructors

4.3: Multiple Constructors in One Class

4.4: Introduction to Access Control

4.5: Other Member Functions

4.6: Static Members

4.7: Finishing Touches

4.8: Any Last Requests?

4.9: Classic Data Structures Using Objects: A Stack

4.10: More Structure: Packages and Nested Classes

4.11: Packages

4.12: Nested Classes

**Module 5: Class Inheritance and Polymorphism**

Reading

* Java in a Nutshell, Chapter 3 (the whole chapter)
* Arnold, Gosling, and Holmes, Chapters 3, 4, and 18

Commentary

5.1: Prologue

5.2: Inheritance

5.3: The protected Access Level

5.4: Abstract Methods and Interfaces

5.5: Polymorphism: More than Just a Twenty-Dollar Word

5.6: Redefining Nonabstract Methods and Extending Concrete Classes

**Module 6: Java Collections Framework**

Reading

* Arnold, Gosling, and Holmes
  + Chapter 8
  + Chapter 2.6.5 (Parameter Values)
  + Chapter 21 introduction, 21.1-21.9 (Collections, iterators)
  + Chapter 10.5.2 (enhanced for statement)
  + Chapter 11 introduction, 11.1 introduction (p. 247-252) (generics)

Commentary

6.1:. Prologue

6.2: Type Wrappers

6.3: Autoboxing

6.4: Java Collections Framework

6.5: Collections

6.6: Maps

6.7: Iteration

6.8: Generics

**Module 7: The Java Language: Advanced Features**

Reading

* Java in a Nutshell, Chapter 5 (pages 238-52)
* Arnold, Gosling, and Holmes, Chapters 12 (sections 12.1-12.7 only) and 14

Commentary

7.1: Prologue

7.2: Exception Handling

7.3: A Few Words about Style

7.4: Multithreaded Programs

7.5: Scheduling

7.6: Synchronization

7.7: Condition Variables

**Module 8: Networking and I/O Programming with Java**

Reading

* Arnold, Gosling, and Holmes, Chapters 20, 23, and 25

Commentary

8.1: Prologue

8.2: I/O and Inheritance in Java

8.3: Chaining Streams

8.4: Dealing with Files in Cross-Platform Code

8.5: Networking

8.6: World Wide Web Access in Java

8.7: Socket Programming in Java

**Module 9: XML with Java**

Reading

* XML for the absolute beginner, JavaWorld
* Java API for XML Processing (JAXP) Tutorial (Chapter 3 Document Object Model)
* Streaming API for XML (StAX)
* Interface XMLStreamConstants

Commentary

9.1:. Prologue

9.2: The XML Document

9.3: The Document Type Definition (DTD)

9.4: Java XML APIs

9.5: XML Parsing with DOM

9.6: Parsing with StAX

9.7: XML Creation using StAX

9.8: Web Services

**Module 10: Miscellaneous APIs**

Reading

* Java in a Nutshell, Chapter 7 (pages 320-25) and Chapter 5 (pages 276-86)

Commentary

10.1: Prologue

10.2:. The Reflection API

10.3: The Class Class

10.4: Fields

10.5: Methods

10.6: JAR Files and Resources

10.7: The Java Beans API

10.8: The Object Serialization API

10.9: The Remote Method Invocation API

10.10: The JDBC API

10.11: XML

10.12: J2EE

**Final Project**