

## Syllabus

**Course Number: CS 434**

**Course Title: Object-Oriented Programming Using Java**

### Course Description:

CS 434 – Object-Oriented Programming Using Java (3). Introduces the Java programming language. Includes control structures, data structures, file input and output, applets, methods, classes and objects, inheritance, polymorphism, multi-threading, graphics and animation.

### Prerequisite Courses:

CS208 Computer Science Fundamentals  
and  
CS362 Data Structures

In order to successfully complete this course, students are expected to have taken the prerequisite CS362 course or its equivalent. From CS362, you should have a working knowledge of the following topics:

1. Practice with problem definition, solution construction and algorithmic development using top-down design techniques.
2. Ability to implement control structures used for sequencing, selection, and iteration.
3. Ability to choose and implement appropriate data structures including arrays, structures, linked lists, and file processing.
4. Understanding of modular code design and how to design test cases.

Although CS434 introduces the Java Programming language, it is also an upper level Computer Science course that introduces advanced programming concepts. Consequently, the ability to program at the CS362 learning outcome level is essential for successful in this Java course.

### Course Overview

This course is an overview of the core features of the Java language. It provides practical, hands-on information that every Java developer should know.

CS434 is challenging and covers a substantial amount of material at a rapid pace. Ultimately, programming is a skill that requires the ability to put designs into practice. The only way to succeed is to practice this skill. Therefore, this course will require a **significant** amount of time each week to complete the programming assignments. If you are not willing or able to spend the necessary time, please reconsider whether this is the correct time to attend this class.

## Course Outcomes

Upon completion of this course, learners should be able to:

1. Design, implement, test, and debug object-oriented programs in Java
2. Apply recognized object-oriented principles to the building of high-quality software & components
3. Design, implement, and debug code that responds to raised exception conditions
4. Create an object-oriented Java application that supports a graphical user interface and event-driven programming
5. Assess the benefits and tradeoffs of using a virtual machine and intermediate language
6. Implement and debug Unified Modeling Language designs in appropriate Java code
7. Design, implement, test, and debug programs that use fundamental control and data structures
8. Design, implement, test, and debug a Java-based layered architecture consisting of Domain objects operated upon within Presentation, Business, and Persistence service layers.

## Course Materials

### *Required Texts:*

Bravaco, R & Simonson, S. (2010) *Java Programming: From the Ground Up*. (1<sup>st</sup>). New York, NY: McGraw-Hill. ISBN: 978-0-07-352335-4.

### *Required Resources:*

Online course Content available via Regis Worldclass at [online.regis.edu](http://online.regis.edu)

### *Technology Tools:*

1. A PC-compatible computer running an MS Windows operating systems (Other operating systems are allowed, but see below.)
2. Current Standard Edition Java Development Kit (JDK) and NetBeans IDE, which are available for free from Oracle at:  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

As with most of Regis learning activities, using various software applications to accomplish assignments requires students to exercise a great deal of responsibility for learning how to successfully operate the software environments. Although other operating systems are allowed, it's the student's responsibility to fix any compatibility issues that might arise.

## Pre-Assignments:

1. Review the programming concepts presented in Chapters 2-5 of the text, which covers concepts presented in CS361 and CS362 (e.g. expressions, data types, variables, selection, and repetition).

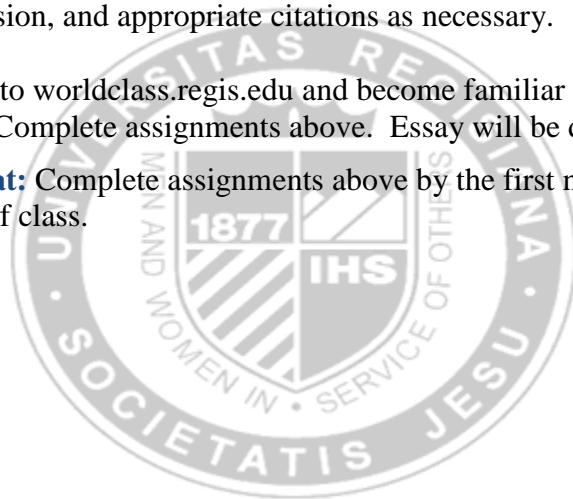
Notice that the Java programming language primarily uses the same C-based syntax found in many contemporary languages, including C++.

The authors have used 'green' highlighting to indicate key concepts and you should familiarize yourself with these concepts. In general, you can ignore concepts that are Java specific since we'll introduce these later in the course.

2. Write a 2 - 3 page paper that compares and contrasts object-oriented programming to traditional structured programming. Keep in mind that you are writing a thesis-support essay that will be based on your own convictions. Begin with an introduction that states the purpose of the paper, your position on the issue, a body that supports and argues your position, a conclusion, and appropriate citations as necessary.

**Online Format:** Sign on to [worldclass.regis.edu](http://worldclass.regis.edu) and become familiar with the course navigation of the Web Curriculum. Complete assignments above. Essay will be due on Friday of week 1.

**Classroom-based Format:** Complete assignments above by the first night of class. Essay will be due on the first night of class.



## Course Assignments and Activities:

	Topics	Readings	Activities Assignments and Associated Points*
1	<ul style="list-style-type: none"> <li>Review: Control Structures &amp; Data Types</li> <li>Java Concepts: <i>Overview &amp; History</i> <i>Classes, Objects, Methods</i> <i>JVM &amp; Byte Code</i></li> <li>Software Engineering: <i>SDLC, Layers, &amp; UML</i></li> </ul>	<p>Online Content for Week 1</p> <p>Textbook: §1.4, 9.1-9.5 10.1-10.3 §6.1-6.3, 10.6 -10.9</p>	<p>Participation in Discussions (5% for the entire course)</p> <p>Week 1: Activity 3 (5%) <i>Programming Project 1: Hello World Program</i></p> <p>Week 1: Activity 4 (5%) <i>Programming Project 2: Use Case Diagram</i></p> <p>Week 1: Activity 5 (3%) <i>Essay: OOP vs Structured Programming</i></p>
2	<ul style="list-style-type: none"> <li>Object-oriented Concepts: <i>Inheritance, overloading, visibility, polymorphism, and constructors</i></li> <li>Java Concepts: <i>Classes, Interfaces, packages</i></li> <li>Software Engineering: <i>Domain layer and testing JUnit</i></li> </ul>	<p>Online Content for Week 2</p> <p>Textbook: §6.4 , 12, 13</p>	<p>Participation in Discussions</p> <p>Week 2: Activity 2 (5%) <i>Programming Project 3: Domain classes and Unit testing</i></p>
3	<ul style="list-style-type: none"> <li>Object-Oriented Concepts: <i>Associations</i></li> <li>Java Concepts: <i>Collections (lists &amp; maps)</i> <i>Arrays, Exception Handling</i></li> </ul>	<p>Online Content for Week 3</p> <p>Textbook: §7, 14, 16, 17</p>	<p>Participation in Discussions</p> <p>Week 3: Activity 2 (10%) <i>Programming Project 4: Collections</i></p>
4	<ul style="list-style-type: none"> <li>Software Engineering (GUI): <i>Model, View, &amp; Controller</i> <i>Presentation &amp; Business layers</i></li> <li>Java Concepts (GUI): <i>AWT and Java Swing</i></li> </ul>	<p>Online Content for Week 4</p> <p>Textbook: §18 &amp; 19</p>	<p>Participation in Discussions</p> <p>Week 4: Activity 2 (10%) <i>Programming Project 5: GUI</i></p> <p>Midterm exam (7%)</p>
5	<ul style="list-style-type: none"> <li>Software Engineering: <i>Persistence Layer</i> <i>Factory Design Pattern</i></li> <li>Java Concepts: <i>Serialization and File IO</i></li> </ul>	<p>Online Content for Week 5</p> <p>Textbook: §15</p>	<p>Participation in Discussions</p> <p>Week 5: Activity 2 (10%) <i>Programming Project 6, Service Creation &amp; IO</i></p>

6	<ul style="list-style-type: none"> <li>• Software Engineering: <i>Persistence Layer</i></li> <li>• Java Concepts: <i>Persistence using JDBC</i></li> </ul>	Online Content for Week 6	Participation in Discussions  Week 6: Activity 2 (10%) <i>Programming Project 7: JDBC</i>
7	<ul style="list-style-type: none"> <li>• Software Engineering: <i>OSI</i></li> <li>• Java Concepts <i>Networking Using Sockets</i></li> </ul>	Online Content for Week 7  Textbook: §15 Bigger Picture (pp. 750-757)	Participation in Discussions  Week 7: Activity 2 (10%) <i>Programming Project 8: Sockets</i>
8	<ul style="list-style-type: none"> <li>• Concurrency &amp; Multi-Threading</li> </ul>	Online Content for Week 8  Textbook: §15 Bigger Picture (pp. 753)	Participation in Discussions  Week 8: Activity 2 (10%) <i>Programming Project 9: Multi-threaded app.</i>  Final Exam (10%)
<b>Total</b>			<b>100%</b>

Online due dates are Sunday, midnight, Mountain Time.

**\*Note to Classroom sections only:** Exact dates for reading assignments and programming assignments may be different than indicated in the above Course Assignments and Activities grid. Your facilitator's syllabus, handed out the first night of class, will indicate any changes.

## Course Policies and Procedures:

### Programming Assignments

Each programming assignment will involve designing, implementing, and testing a program using the concepts discussed in the book and class. **Note:** programs that do not compile, are not modular, nor properly documented using Javadoc standards will not be accepted.

Programming assignments will be graded according to details listed in the rubrics given in the class, which also includes the following criteria:

1. Source Code
  - a. Code is easy to read and self-documenting
  - b. Appropriate Javadoc comments are used
  - c. Inline commenting is used where appropriate
2. Coding Style
  - a. Program is modular
  - b. The Java programming language is used correctly
3. Functionality
  - a. There are no compilation, logic, and design errors.
  - b. The program solves the assigned problem.
4. Testing
5. Analysis (if required)
  - a. Logical and complete

## Exams

There will be a midterm and final exam. Exams questions will be cumulative, taken from reading assignments, programming assignments, and class participation. Exams may test your understanding, comprehension, application, and evaluation of the material presented in this class.

## Participation

Class participation/effort is important because we can all learn from each other. Your participation points can make a difference in the final grade. Participation means:

1. Present in class every session (classroom)  
Present in the forum every week (online)
2. Effectively responds to questions from the facilitator (classroom)  
Regularly checks forum and posts all required items by the deadlines (online)
3. Interacts/replies to other students in classroom/forum discussions.

## CC&IS Grading Scale

Letter Grade	Percentage	Grade Point
A	93 to 100	4.00
A-	90 to less than 93	3.67
B+	87 to less than 90	3.33
B	83 to less than 87	3.00
B-	80 to less than 83	2.67
C+	77 to less than 80	2.33
C	73 to less than 77	2.00
C-	70 to less than 73	1.67
D+	67 to less than 70	1.33
D	63 to less than 67	1.00
D-	60 to less than 63	.67
F	Less than 60	0

*Additional information about grading can be found in the latest edition of the University Catalog, available at <http://www.regis.edu/Academics/Course%20Catalog.aspx>.*

## CC&IS Policies and Procedures

Each of the following CC&IS Policies & Procedures is incorporated here by reference. Students are expected to review this information each term, and agree to the policies and procedures as identified here and specified in the latest edition of the University Catalog, available at <http://www.regis.edu/Academics/Course%20Catalog.aspx> or at the link provided.

- The CC&IS Academic Integrity Policy.

- The Student Honor Code and Student Standards of Conduct.
- Incomplete Grade Policy, Pass / No Pass Grades, Grade Reports.
- The Information Privacy policy and FERPA. For more information regarding FERPA, visit the [U.S. Department of Education](http://www.ed.gov).
- The HIPAA policies for protected health information. The complete Regis University HIPAA Privacy & Security policy can be found here: <http://www.regis.edu/About-Regis-University/University-Offices-and-Services/Auxiliary-Business/HIPAA.aspx>.
- The Human Subjects Institutional Review Board (IRB) procedures. More information about the IRB and its processes can be found here: <http://regis.edu/Academics/Academic-Grants/Proposals/Regis-Information/IRB.aspx>.

The CC&IS Policies & Procedures Syllabus Addendum summarizes additional important policies including, Diversity, Equal Access, Disability Services, and Attendance & Participation that apply to every course offered by the College of Computer & Information Sciences at Regis University.

A copy of the CC&IS Policies & Procedures Syllabus Addendum can be found here: <https://in2.regis.edu/sites/ccis/policies/Repository/CCIS%20Syllabus%20Addendum.docx>.

