

# CSC 211 - Syllabus

---

**Instructor:** Deborah Mathews ([deborah\\_mathews@my.uri.edu](mailto:deborah_mathews@my.uri.edu))

**Text:** Frank M Carrano,  
*Imagine! Java: Programming Concepts in Context*,  
Prentice Hall, 2011.

**Objectives:** CSC 211 is intended for majors in Computer Science and Computer Engineering (and Mathematics, Statistical Science, and Physics). CSC 211 is designed to provide students with an introduction to object-oriented programming and principles of good design, algorithm development, data representation and manipulation.

**Prerequisites:** Prior experience with computers and programming, and [MTH 111 - Precalculus](#), or equivalent.

**Software:** We will use the **Eclipse** programming environment in this course. Eclipse is a piece of freeware, and can be downloaded as a zip file (look under the **Miscellaneous** tab).

**Assignments, Examinations, Grading:** Reading and questions on the reading will be due the day before each class. Four programming assignments and a final project will be assigned to be written using Java. Laboratory assignments will be assigned once a week. Two mid-term examinations and a final examination will be administered during the semester (see the [CSC 211 home page](#) for the exact dates). Quizzes will be administered once a week, with the exception of the weeks when there is an exam. The final grade will be based on the individual grades received on the lab assignments, the reading questions, the programs, the quizzes, the exams, and the final. The following coefficients (or weights) will be used in the determination of the final grades:

Lab assignments:	10%
Twice-weekly questions:	10%
Programming assignments:	35%
Quizzes:	10%
Examinations:	20%
Final examination:	15%

**Late Submission of Assignments:** All assignments must be turned in the day they are due **by the time listed on Sakai**. If an assignment is not turned in on time, 10% will be taken off for each late day (or any fraction of a day). Unless otherwise specified, lab assignments will be done during the lab and due at the end of the lab period.

**Plagiarism, Cheating:** While you are welcome to talk with others about your programs, you may not work together on your assignment with anyone else. Your program must be your own work. If you get help from another student on any part of your assignment, be sure to acknowledge that student in your documentation. If you find code on the web or from some other source that you use or model your code after you must cite your source. Unauthorized cooperation or any form of cheating will be brought to the attention of the Dean for disciplinary action. See the appropriate sections of the University Manual.

**Laboratory Use, Software Piracy:** Software piracy will be dealt with exactly like vandalism or stealing of university or departmental property. Any abuse of computer or software equipment will be brought to the attention of the appropriate authority for disciplinary action.

**Cell phones:** Please turn your cell phone off (or put it on vibrating mode) before the beginning of the class.

**Topics:** The following topics will be covered during the semester, although not necessarily in this order.

1. **Computers and Programming**

- computer terminology
- problem-solving, algorithms, programs, and languages

2. **Basic Elements of Java**

- identifiers, variables, and constants
- expressions and assignment statements
- data types
- comments
- structure of a Java program
- simple input/output

3. **Flow of control**

- **if** and **switch** statements
- logical operators, **Boolean** expressions
- **for**, **while**, and **do-while** statements
- nesting of loops
- choosing an appropriate loop structure

4. **Classes, and objects**

- class definitions, methods, instance variables
- objects
- methods and parameters
- constructors
- overloading
- class hierarchy, object relationship

5. **Software design**

- information hiding
- top-down program development
- divide-and-conquer algorithms

6. **Inheritance**

- software re-use, derived classes
- polymorphism, overriding

7. **Recursion**

- recurrence relations
- simple recursion, indirect recursion
- recursion vs. iteration

8. **Basic Data Structures**

- arrays
- vectors

9. **Exception and Input/Output**

- exception handling
- using exception
- I/O streams, file processing