### AP Computer Science A: COSC 431-1

#### Instructor Information:

<table>
<thead>
<tr>
<th>Name: Michael Lackner</th>
<th>Email: <a href="mailto:mlackner@blakefield.loyola.edu">mlackner@blakefield.loyola.edu</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone Extension: 443-841-3320</td>
<td>Office Hours: By Appointment Only</td>
</tr>
</tbody>
</table>

#### Course Description

This class covers the Java programming language with special emphasis on preparing for the Computer Science A exam currently scheduled for **Tuesday, May 6, 2008**. The class will cover all of the topics needed to prepare for the A exam. **To successfully complete the class, all students must take the APCS A exam.**

#### Course Objectives:

- Understand the main principles of object-oriented software design and programming
- Learn to code fluently in Java in a well-structured fashion and in good style; learn to pay attention to code clarity and documentation
- Learn to use Java library packages and classes within the scope of the AP/A Java subset
- Understand the concept of an algorithm; implement algorithms in Java using conditional and iterative control structures and recursion
- Learn common searching and sorting algorithms: Sequential Search and Binary Search; Selection Sort, Insertion Sort, and Mergesort
- Understand arrays and the **ArrayList** class
- Acquire skills in designing object-oriented software solutions to problems from various application areas
- Learn the GridWorld case study and accompanying exercises and questions provided by the College Board
- Discuss ethical and social issues related to the use of computers
- Prepare for the A-level AP exam in computer science.

#### Texts and Supplementary Materials:


The College Board’s **GridWorld** case study.
Syllabus: AP Computer Science A


Introduction to Programming Using Java, Fifth Edition (online textbook) Author: David J. Eck (eck@hws.edu) [http://math.hws.edu/javanotes/](http://math.hws.edu/javanotes/)

Current magazine and Internet articles discussing ethical and social issues related to computer use.


**Course Outline:**

Chapter numbers for readings and exercises refer to *Java Methods A&AB, AP Edition.* The labs, case studies, and projects proposed below come from *Java Methods* and serve only as examples of possible assignments; the teacher’s favorites may be used instead. *GridWorld* refers to the College Board’s *GridWorld* case study narrative.

**Unit 1: Review of the Introductory Material (4 weeks)**

**Introduction to Hardware, Software, and the Internet / Review of Java syntax and style**
(Weeks 1-2; duration 2 weeks)

**Introduction to Hardware, Software, and the Internet**

Elements of a computer system. How information is represented in computer memory. A glimpse of binary and hex systems and ASCII / Unicode.

**Software development environment (Week 2; duration 1 week)**

Getting familiar with the software development process. Compilers and interpreters. JDK tools (*javac, java, appletviewer, javadoc*). Running a Java program in a command-line environment (optional). Using an IDE. Java classes and source files. A brief introduction to OOP.


*Reading and exercises: Chapters 1, 5-7, Appendix A. Introduction to Programming Using Java (online textbook) Chapter 1*
Section 1: The Fetch and Execute Cycle: Machine Language
Section 2: Asynchronous Events: Polling Loops and Interrupts
Section 3: The Java Virtual Machine
Section 4: Fundamental Building Blocks of Programs
Section 5: Objects and Object-oriented Programming
Section 6: The Modern User Interface
Section 7: The Internet and the World-Wide Web

Labs: Exercises and/or free-response questions from the Test Package for chapters 5-7.

1. Review of Algorithms (Week 3; duration 1 week)

Algorithms using iterations and recursion.

Reading and exercises: Chapters 4 and 8.
Labs: Exercises and/or free-response questions from the Test Package for chapters 4 and 8.

2. Review of classes and objects (Week 4; duration 1 week)

Classes and objects. Fields, constructors and methods. An introduction to inheritance. First Steps case study review.

Reading and exercises: Chapter 3.
Lab: “Bystander” in the First Steps (Exercise 3-12).

Unit 2: Classes, Class hierarchies, GridWorld (7 weeks)

3. An introduction to GridWorld (Week 5; duration 1 week)

Experimenting with the GridWorld GUI. An overview of the classes and objects involved. Role-play exercise.

Reading and exercises: GridWorld Part 1.
Lab: Set up a GridWorld project and run BugRunner.
Lab: Simple extensions of the Bug class.
4. **Details of defining classes and using objects (Weeks 6-7; duration 2 weeks)**


*Reading and exercises:* Chapter 9; GridWorld Part 2.
*Lab:* Snack Bar.
*Lab:* GridWorld exercises for Part 2 (page 12).

5. **Strings (Week 8; duration 1 week)**


*Reading and exercises:* Chapter 10.
*Lab:* Lipograms (Section 10.8).

6. **Class hierarchies, abstract classes, and interfaces (Weeks 9-11; duration 3 weeks)**


*Reading and exercises:* Chapter 11, GridWorld Part 3.
*Lab:* Dance Studio.
*Lab:* Past free-response questions on class hierarchies and polymorphism.
*Lab:* Creating a subclass of `Actor`, GridWorld Part 3 group activity (p. 24).

**Unit 3: Arrays, ArrayLists, searching and sorting (8 weeks)**

7. **Arrays (Week 12; duration 1 week)**


*Reading and exercises:* Sections 12.1-12.3.
*Lab:* Fortune Teller (Section 12.3).
8. **ArrayLists.** Array and **ArrayList** algorithms (Weeks 13-14; duration 2 weeks)

- **List** interface. **ArrayList**’s constructors and methods. Traversals and the “for each” loop. Finding the largest and the smallest element. Inserting and removing elements. **ArrayList**s in GridWorld.

  *Reading and exercises:* Sections 12.4-12.9, GridWorld Part 4.
  *Lab:* Creating an Index for a Document (Section 12.9).
  *Lab:* Past free-response questions on arrays and **ArrayList**.
  *Lab:* GridWorld Part 4 exercises (p. 32).

9. **Two-dimensional arrays (Weeks 15-16; duration 2 weeks)**

Declaring 2-D arrays. Indices. Accessing the number of rows and columns. Traversals.

  *Reading and exercises:* Sections 12.10-12.11.
  *Lab:* Chomp (Section 12.11).

10. **Searching and Sorting (Weeks 17-19; duration 3 weeks)**

Comparing objects. The **equals** method and the **Comparable** interface. Sequential and Binary Search. Selection Sort, Insertion Sort, and Mergesort. The **java.util.Random** class.

  *Reading and exercises:* Chapter 13.
  *Lab:* Chapter 13 exercises (e.g., 13-4, 13-9).
  *Lab:* Keeping Things in order (Section 13.4).
  *Lab:* Benchmarks (Section 13.9).

**Unit 4: Enrichment (5 weeks)**

11. **Streams and files (Weeks 20-21; duration 2 weeks)**

Text and binary files. Streams vs. random-access files. Java I/O package. The **Scanner** class. Checked exceptions.

  *Reading and exercises:* Chapter 14.
  *Lab:* Dictionary for Ramblecs (Section 14.5).
  *Lab:* Exercises and projects from the Test Package for Chapter 14.

12. **Graphics and GUI (Weeks 22-24; duration 3 weeks)**

Computer graphics concepts. The Java **Graphics** class. GUI components and their events. Layouts. Handling mouse and keyboard events.
Reading and exercises: Chapters 15, 16, 17.
Lab: Pieces of the Puzzle (Section 15.7).
Programming project: Ramblecs (Section 16.6).
Programming project: Drawing Editor (Section 17.4).

Unit 5: Review (6 weeks)

13. GridWorld review (Weeks 25-27; duration 3 weeks)

Review of the GridWorld classes and interfaces. Modifications and exercises.

Reading and exercises: GridWorld Parts 1-4; Be Prepared Chapter 7.
Labs: GridWorld Enhancements (from suggested exercises for Part 4, p. 32, and Be Prepared).

14. Review and Practice for the Exam (Weeks 28-30; duration 3 weeks)

Reading: Be Prepared Chapters 1-5; Be Prepared Chapter 8 (past free-response questions and solutions), Be Prepared practice exams, 175 Multiple-Choice Questions in Java

Grading:
Tests:
There will be several major tests per quarter on the Computer Science testing days (A & F).

Quizzes:
Given periodically (announced and un-announced) to reinforce concepts and mark progress.

Homework:
Homework may be assigned and should be completed in time for the next day’s class period. Homework checks will be frequent and random.

Programming:
The bulk of your work will be actual Java programming. Programming projects, which are due IN CLASS, on the due date. Late projects will be accepted but the grade will be reduced one full letter grade per day late. The programming assignments will be graded in several different areas, not just whether or not the programs “work.” These areas include, but are not limited to: understandability, adaptability, and reusability. The ramifications of these issues will be discussed in class.

Quarter/Semester/Final grades will be determined by a straight points system:
Earned Pts/ Available Points

The following categories will be used to determine grades:
Projects, Tests/Quizzes, Homework, Class Participation*

*Note: Class Participation will be determined by the following criteria: collaboration, communication, information literacy, creativity