Instructor: Gerry Jenkins, Jim Mackowiak  
Semester: Spring 2009  
Section: 36129  
Session Dates: January 12th to May 27th, 2009  
Emails: Gerry Jenkins gjenkins@lbcc.edu  
Phone: (562) 938-4623 Jim Mackowiak jimmacksmail@yahoo.com  
Office Hours: see gjenkins.lbcc.edu

Course Description
Course Number: CS 21  
Units: 3.5  
Recommended Preparation:  
Programming (CBIS 7) and Trigonometry (MATH 40)  
This is the first of three introductory courses in computer science. This course  
introduces the basics of computer software design, programming, and related  
underlying concepts required with java programming. You will also learn a bit about  
object-oriented design, a fair amount about object oriented programming, using  
Java as the “language of practice.”

Course Student Learning Outcomes
Upon completion of the course the student should be able to:

PROGRAMMING FUNDAMENTALS:  
Design programs in java, showing a mastery of using the following language  
features: basic statements (assignment, I/O); basic data types (int, long,  
double, boolean); selection (if, switch); loops and loop control (for, while, do);  
methods/functions (defining, calling, parameter passing).  
Create programs in Java that show a proficiency in using exception handling as  
applied to files, and scoping rules.  
Recognize the following features in a program: the float and char data types,  
exception handling in general, simple recursion, pointers, applets, event-driven,  
programming, and a graphical user interface.

DATA STRUCTURES:  
Design programs in java, showing a mastery of the following data structures:  
strings, single-dimension array; and singly-linked lists with an iterator.  
Create programs in Java that show a proficiency in the use of sequential text  
files and java Array Lists.  
Recognize the following features in a program: Java Vectors, parallel arrays  
multi-dimension arrays and heterogeneous nested data structures.

CONVENTIONAL ALGORITHMS:  
Design programs in java, showing a mastery in the following topics: linear  
search, computing the sum, average, minimum, and maximum; traversing an  
array, array list or singly-linked list and applying an operation to each element;
opening, reading from, writing to, closing and handling errors associated with sequential text files; moving within, adding items to and deleting items from a list, implemented as an array, an arraylist, or a singly-linked list.

**OBJECT ORIENTATION:**
Develop programs that show a mastery in the extending and using existing classes, and is able make simple modifications to class definitions and to create and use instances of classes appropriately.

Design programs that use function overloading, and can design a simple new class with constructor(s), methods, and data members, and use predefined class families.

Recognize the following features in a program: recognize the importance of the OO paradigm, design an object-oriented solution to a simple problem, extend a straightforward object-based design, gain a basic conceptual understanding of generic types, and gain a basic conceptual understanding of inheritance.

**PROGRAM DESIGN AND IMPLEMENTATION:**
Demonstrate mastery in the process of entering, running, modifying code using Java tools.

Create a program, given a prose specification and possibly the description of available library routines, applying the mastery and proficiency level topics listed above as appropriate to solve a new problem in a familiar domain.

Create programs in Java that show a proficiency in following specified coding style guidelines, appreciates the value of coding style guidelines, can locate and correct simple errors in code, and develop simple test cases.

**PROGRAMMING SKILLS:**
Design programs in java, showing a mastery to design and implement methods for a class, design and implement static methods, design and implement a complete program that uses an existing class definition, debug a program with the help of 'print' statements and thoroughly test a small program.

Plan and create a complete program that uses two existing class definitions, where an object of one class uses information provided by an object of the other class.

Demonstrate a basic understanding of unit testing.

---

**Course Content and Delivery**

<table>
<thead>
<tr>
<th>Meeting times and places</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab</td>
<td>Monday's</td>
</tr>
<tr>
<td>Lecture</td>
<td>Wednesday's</td>
</tr>
</tbody>
</table>

This class is a web enabled class with a required website located at http://cbisdl.lbcc.edu/spring
This site may be used to post materials, schedules, discussions, and grade results for the class.
This site may also be used for the deliver of various assessment parts of this class.

**Textbook and Technical requirements**
The required textbook for this class is:

"JAVA Programming, Guided Learning with Early Objects" by D.S. Malik and Robert Burton
Published by Course Technology ISBN: 1-4239-0162-2
You can purchase this book at our LBCC LAC book store. You will need the book the first week of class.
We will be using the current Java SDK with a text editor called TextPad.

**Grading**
Your final grade will be composed based on the sum of these assessment types weighted as indicated:

<table>
<thead>
<tr>
<th>Assessment type</th>
<th>Percent of Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab exams (7 to 9 of them)</td>
<td>40%</td>
</tr>
<tr>
<td>Lecture Exams (3)</td>
<td>30%</td>
</tr>
<tr>
<td>Assignments</td>
<td>20%</td>
</tr>
<tr>
<td>Lab and Lecture In Class Participation (attendance, and assignments to be done in the lab or lecture)</td>
<td>10%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

The final grade will be awarded according to a standard scale:

<table>
<thead>
<tr>
<th>Total percent</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greater than or equal to 90%</td>
<td>A</td>
</tr>
<tr>
<td>Greater than or equal to 80%</td>
<td>B</td>
</tr>
<tr>
<td>Greater than or equal to 70%</td>
<td>C</td>
</tr>
<tr>
<td>Greater than or equal to 60%</td>
<td>D</td>
</tr>
<tr>
<td>Below 60%</td>
<td>F</td>
</tr>
</tbody>
</table>

**Course Policies**

**Due Dates:** Each assignment will be given a due date, the time due will always be by the end of the lab or lecture meeting regardless of what the moodle site says.

For late assignments, 25% of the possible points will be deducted for the first two weeks or part of a week the assignment is late. Assignments that are turned in more than two weeks late will be given zero points. Assignments where the answer is given on the due date have no points if turned in after the answer is given. In Class work is due on the date it is assigned and cannot be done later. Lab test can be retried once only.

All assignments must be in on or before the last date of class.

**Cheating:**
You can not use someone else's work or copy from any source. This includes using the same file from a friend, relative, or anyone else. Anyone participating in any form of academic dishonesty will be withdrawn from the class and your case forwarded to the Dean of Student Affairs for consideration of any other actions. This includes the originator of any work that is copied. You are responsible to protect your work. If it is stolen, you have to report it immediately to your instructor.
Please contact the Student Affairs office to be informed of your rights. Cheating includes: copying any part of a document that is not yours, including copy and paste from the Internet, using any communication device or method in a test, doing someone else's work in part or whole, plagiarism in any form, and using any form of note in a closed book test. Do not let any student use your files, media or computer. You may be responsible for any material they use from you.

**Withdrawal from class:**
If you are enrolled in the class at the start of the semester but never contact the instructor and participate in the class by the census date of the class (see the deadlines at [http://admissions.lbcc.edu](http://admissions.lbcc.edu)), you will be considered a "No Show" and I will drop you from the class. **In all other circumstances, if you need to withdraw from this class, you are responsible for dropping the class through Admissions and Records or online.** If for some reason you decide to drop the class early in the semester and do not wish to receive a "W" grade on your transcript, you must go to the Admissions and Records Office and drop the class by date indicated as the last day to drop and receive a "W" (see the deadlines at [http://admissions.lbcc.edu](http://admissions.lbcc.edu)). (You can also drop the class online.) If you have not dropped the class as of this date, you will be assigned a letter grade (A, B, C, D, F, Cr or NC) in the course, whether or not you are attending class.

**PLEASE NOTE THAT IF YOU NEED TO WITHDRAW FROM THIS CLASS, YOU ARE RESPONSIBLE FOR DROPPING THE CLASS THROUGH ADMISSIONS AND RECORDS OR ONLINE.** If you simply stop participating in class and do not drop the class, you will be kept on the roll book and you will be assigned a grade in the class.

**Disclaimer:**
This syllabus is subject to change during the course of the class. Students will be notified of any change.

_Last modified: Thursday, 8 January 2009, 01:15 PM_