ALFRED UNIVERSITY
COLLEGE OF BUSINESS
MANAGEMENT INFORMATION SYSTEMS

Course Syllabus

Course:	MIS 290 – Computer Programming – Java Cisco Networking Academy™ Sponsored Program Curriculum	
Date:	Spring 2003	
Instructor:	Prof. Clare Popejoy	Office: Olin 433
E-mail:	popejoy@alfred.edu	Phone: 607-871-2829

Office Hours: MWF 1:00 – 2:00 pm
Other times by appointment

COURSE DESCRIPTION:
An introduction to the fundamentals of computer programming. The course is offered using Visual Basic and Java in alternate years, and may be repeated once for credit. Topics covered include programming concepts, program design and development, debugging and testing.

COURSE PREREQUISITES: MIS 101 or equivalent.

COURSE OBJECTIVES:

- Understanding the history and current use of object-oriented programming and the Java language to solve business problems.
- Knowledge of Java environment and use of Java development kit.
- Application of Java language keywords and syntax to create statements for declaring and storing java data types.
- Ability to write code that implements object-oriented principles and design encapsulation, composition, and inheritance.
- Upon completion of course, students will receive a Fundamentals of Java Programming Certificate of Completion.
- Upon completion of course, students will receive preparation for the Sun Certified Programmer for Java 2 Platform certification exam.

REQUIRED TEXT: *None*

SUPPORTING WEB SITES:

Blackboard site:	http://blackboard.alfred.edu/
Alfred's Java site:	http://netacad.alfred.edu/java (for curriculum)
Sun's Java site:	http://java.sun.com/ , then APIs, under documentation
Cisco Academy site:	http://cisco.netacad.net/ (for curriculum and exams)
Blue J	http://www.bluej.org (we are using version 1.20)

COURSE SUPPLIES:

- At least three (3) blank diskettes (1 work disk; 1 turn-in disk; 1 backup disk) [*Optional*: zip disks]

TENTATIVE COURSE SCHEDULE – MIS 290 – Spring 2003 – rev. 1/23/03			
Week	**Dates**	**Class Activities**	**Assignments/Readings Due**
1	1/22/03	Introduction and course overview	
	1/24/03	Object Oriented Programming & Java	Ch 1 & 2
2	1/27/03	Java Language Elements	Ch 3
	1/29/03		
	1/31/03	**Exams 1 & 2**	
3	2/03/03	Java Language Operators & Control Structure	Ch 4
	2/05/03		
	2/07/03	**Exam 3 & 4**	
4	2/10/03	Basics of Defining & Using Classes	Ch 5
	2/12/03		**Due**: Ch 1- 4 labs (individual)
	2/14/03	**Exam 5**	
5	2/17/03	System, String, StringBuffer, Math & Wrapper Classes	Ch 6
	2/19/03		**Due**: JBANK Phase I
	2/21/03	**Exam 6**	
6	2/24/03	Arrays	Ch 7
	2/26/03		
	2/28/03	**Exam 7**	
7	3/03/03	Classes & Inheritance	Ch 8
	3/05/03		**Due**: JBANK Phase II
	3/07/03	**Exam 8**	
8	3/10/03 – 3/14/03	**SPRING BREAK**	
9	3/17/03	Understanding Packages	Ch 9
	3/19/03		**Due**: JBANK Phase III
	3/21/03	**Exam 9**	
10	3/24/03	Creating GUIs using AWT	Ch 10
	3/26/03		**Due:** Ch 6-9 labs (individual)
	3/28/03	**Exam 10**	
11	3/31/03	Applets & Graphics	Ch 11
	4/02/03		
	4/04/03	**Exam 11**	
12	4/07/03	Exceptions	Ch 12
	4/09/03		**Due**: JBANK Phase IV
	4/11/03	**Exam 12**	
13	4/14/03	Files, Streams, Inputs & Outputs	Ch 13
	4/16/03		
	4/18/03	**Exam 13**	

TENTATIVE COURSE SCHEDULE – MIS 290 – Spring 2003 – rev. 1/23/03			
Week	**Dates**	**Class Activities**	**Assignments/Readings Due**
14	4/21/03	Collections	Ch 14
	4/23/03		**Due**: Ch 10-13 labs (individual)
	4/25/03	**Exam 14**	
15	4/28/03	Threads	Ch 15
	4/30/03		
	5/02/03	**Exam 15**	
16	5/05/03	Review for final exam	**Due**: JBANK Phase V
	5/09/03	**Final Exam** -- 1:15 – 3:15 pm	

COURSE POLICIES AND REQUIREMENTS:

Course Schedule

The student is responsible for knowing announced quiz, exam, and due dates on the Course Schedule or as changed in class. This Course Schedule is tentative and may be adjusted to meet the needs of a particular class. The professor reserves the right to make adjustments and changes in this syllabus at any class session. Changes will be announced in class and posted in writing to Blackboard. No due date for a programming assignment will be changed to a date earlier in the semester.

Class Attendance

A student who is absent from part or all of a class session is still responsible for assignment due date, the material covered in class, and for any changes announced to class activities, class schedule, or project requirements. Students who are unable to attend class on a due date may turn in the assignment early to avoid late penalties.

Electronic Contact with Professor

From time to time students will need to be in contact with the professor. The professor will use Blackboard and e-mail. The professor's e-mail address is listed at the beginning of this syllabus. In the SUBJECT line, students **must** put the course number (MIS 290) plus any brief description of the subject of the correspondence. Students should check Blackboard and their e-mail at least once a week. Do **not** e-mail assignments.

Students can access Blackboard anywhere they have web access. The URL is http://blackboard.alfred.edu, then follow the link to this class. An alternative method is to go to the Alfred University home page, the GoTo box, and select Blackboard 6.0.

Electronic Contact with Others

Students should turn off or set on inaudible their cell phones, pagers and any other electronic devices before coming to class. An audible electronic device is distracting to everyone. Any student whose

electronic device sounds an audible alert in class can expect points to be deducted from his/her participation grade.

Exams

There are Chapter exams throughout the course (see Course Schedule). They are available online from the Cisco Academy web site for a limited period of time. These chapter exams will be "open" exams. In other words, you may use the curriculum, your notes, your lab exercises, etc. At the end of the course there will be a final exam. It will also be available online from the Cisco Academy web site for a limited period of time. The final exam will be a "closed" exam, for which you may not use the curriculum, notes, etc. All of the exams, unless you are otherwise informed, will be taken in class.

Programming Assignments

There are several programming lab exercises in this class, which will be individual. In addition, there is a continuing Case (JBANK) throughout the course, which will be a group project. More details about both kinds of assignments will be posted to Blackboard within a few days of the beginning of the semester. Miscellaneous drill assignments may be assigned as needed. Programming assignments are due at the ***beginning*** of the class meeting on the specified due dates (see Tentative Course Schedule). ***Do not miss class to complete an assignment. Do not work on assignments during classroom presentations by the professor or another student. You will miss valuable tips.*** An initial 5% penalty may be deducted from assignments turned in after the beginning of the class or later than day. An additional 10% penalty may be deducted from assignments turned in the next day and for each additional week the project is late.

Students will adhere to the highest professional and ethical standards, which are necessary to foster both professional growth and personal learning. Plagiarism in any form will result in a lower grade for a programming assignment. Plagiarism may also result in further penalties, at the professor's discretion.

Appropriating code written by current or former students and representing this code as a student's own work is considered plagiarism. Working so closely with another person or tutor on individual programming assignments so that the respective coding solutions are unduly similar is also unacceptable.

Use of coding examples from the Sun's Java site is acceptable and is not considered plagiarism. If you use code from other sources, be sure it is open code and include comment statements that cite the source of the code and explain it completely.

Computer Policies

This class will use the computer network in the AU labs. Students must follow the computer lab rules and policies. They should expect to spend some computer lab time outside class. At a minimum, they should check their e-mail once a week.

Students are required to use their AU email account for this class. It is their responsibility to maintain the security of their computer account and password. Students should change their initial password as soon as possible, at least within one week of account assignments. Students should not allow anyone

else to use their account or to learn their password. Students are encouraged to change their passwords at least once a month or immediately if they think anyone else may have acquired their password.

Students are required to use Blackboard, an electronic tool for which they will have access. Copies of the syllabus and other course information will be posted there. Periodic announcement may also be posted. Students should check the Announcements section at least once a week (preferably before each class meeting). Announcements could include notices of changes in assignments and/or assignment instructions. Students should always check this area before submitting projects.

The Java Curriculum is available anywhere you have Internet access to the netacad.alfred.edu web site. Unfortunately students are not able to download the curriculum in order to work on a local computer.

You can download a copy of Java and of Blue J (see page of this syllabus for the URLs) in order to work in an environment other than the computer lab. Be aware that different versions may act differently and that when I grade assignments, I will probably be using the same version that is installed in the lab.

Students may use computing facilities other than the AU labs as long as they use the supported software packages. They should keep in mind that they may not be able to get technical assistance from their professor or lab aides in using hardware or software not available in the labs.

Grading Policy
Grades will be determined as a percentage of 1,000 points on the following basis:

Assignment	**Percentage**
Programming exercises	10%
Group Project	40%
Exams	20%
Final Exam	30%
Miscellaneous (attendance, participation/ contribution to class meetings, ad-hoc assignments, etc)	10%

Programming Assignments

Individual assignments

- The individual assignments are all the lab activities (except JBANK) in each chapter.
- You will keep all the files on a disk. [be sure to keep adequate backups]
- Name lab files with appropriate names according to the lab instructions.
- File all files for each lab in the appropriately named folder.
- For labs that have questions (they are usually fill in the blank):
    - Create a word processed document
    - Name the file with the appropriate lab name
    - Include the step number (if there is one), the question, and your answer
    - Save the file in the same lab folder with the other lab files
- Put your name, date, and lab number in comment statements in each class of each lab.
- Submit labs according to the time schedule located elsewhere in this syllabus.
- Use comments liberally throughout the code.

Group Project

- The group project is the JBANK project that is a continuing case throughout the curriculum.
- Groups should be 2-4 members. Submit group names and a list of members to the Discussion Board in Blackboard.
- All group files will be submitted on a single disk (one disk per group). [Be sure to keep adequate backups.]
- Each group is encouraged to use Blackboard's Group area for sharing files.
- Use the project instructions for naming files and folders. Store the files in the appropriate folders.
- Any part of the project that has questions to be answered, use the same instructions as for the individual assignments.
- Be sure to include UML models where appropriate.
- Use comments liberally throughout the code.
- JBANK binder (with tabs)
    - Each group will submit a binder with the following:
        - Printout of code for each lab
        - Printout of html pages for each phase
        - Printout of all UMLs
        - Printout of all questions & answers
        - A disk with all project files
    - The binder will be submitted on the due date for each Phase of the project.

- Grading for the group project will be based on the following:
    - Documentation (30%)
        - Submit html files for each phase
            - Documentation of methods and public attributes
    - Coding conventions (10%)
        - Use of indentation
        - Proper naming of variables and methods
        - Inclusion of comments (not javadoc comments)
    - Code works (50%)
        - Error free execution of all classes in the phase
    - Coding enhancements (10%)
        - Special use of the language
        - Incorporation of appropriate core classes
        - Use of professional level user interfaces (e.g. user prompts, proper tabs, headings for output)

JBANK labs matrix

Phase	Labs	Lab Title	Due Date
I	2.6.1	Designing & Describing Classes using UML	2/19/03
	2.6.1.2	Develop the Bank classes for Phase I	
	3.1.6.1	Inserting Documentation for the Classes in the Banking Application	
	3.1.6.2	Generating API Docs for JBANK Classes using the Javadoc Tool	
	3.7.1	Creating the Classes for Phase I of the JBANK Application	
	5.6.4	Constructors and Methods	
	5.11.1	Completing the JBANK Phase I Application	
II	6.9.1.1	System, String, StringBuffer and Use of Console Class	3/05/03
	6.9.1.2	Wrapper Classes, Math Class, Date Class	
	7.6.1	Implementing Arrays in the JBANK Application	
III	8.7.1	Implement Abstraction in Phase II of the Banking Application	3/19/03
	8.7.2.1	Implement Inheritance, Extending from Abstract and Concrete Classes	
	8.7.2.2	Abstraction at Several Levels – Checking Account	
	8.10.1	Polymorphism in the Banking Application	

Phase	Labs	Lab Title	Due Date
IV	9.4.2	Building a Banking Package	4/09/03
	9.6.1.1	Designing a GUI to Represent an ATM for Customers	
	9.6.1.2	Designing the GUI Interfaces	
	10.2.2.1	Creating the Components (TellerView Class)	
	10.2.3.1	Selecting Containers (TellerGUI Class)	
	10.2.3.2	Selecting Containers (ATMGUI Class)	
	10.2.5.1	Layout Managers (TellerGUI Class)	
	10.2.5.2	Layout Managers and Adding Components (ATMGUI Class)	
	10.3.2.1	Identifying Even Handler Features in the TellerGUI Class	
	10.3.2.2	Implement Even Handling for the ATMGUI Class	
	10.3.4	Implement the Model for the ATMGUI Class	
	10.4.1	Finalizing the Model View Controller pattern for the ATMGUI Class	
	11.8.1	Creating the ATM Applet	
V	12.9.1	Exceptions for the JBANK Application	5/05/03
	13.5.1.1	Writing Customer Objects to a File	
	13.5.1.2	Reading Customer Objects from a file	
	14.4.1.1	File I/O using Collection Classes	
	14.4.1.2	Sets and Iterators	