

**Course Syllabus**  
**4002-219**

**Programming for Information Technology III**  
**Winter (20072)**

**REMINDER:** The information presented in this syllabus is subject to expansion, change, or modification during the quarter

<b>Instructor:</b> Jai W. Kang Office: 70-2651 ☎ 475 -5362 ✉ email address: jai.kang@rit.edu	<b>Office Hours:</b> T 4pm – 6pm W 1pm – 3pm Others by appt.
<b>Course Web Page:</b> <a href="http://www.jkang.com">http://www.jkang.com</a>	<b>We will also use a MyCourses conference</b>

**Course Text and Materials**

"Introduction to Java Programming", 6th edition by Y. Daniel Liang, Prentice Hall, 2007, ISBN 0-13-222158-6

**Important RIT Deadlines**

**Last day of drop/add: December 10, 2007**

**Last day to withdraw from the course with a “W”: February 8, 2008.** The deadline for withdrawing from a course with a W grade is the end of the 8<sup>th</sup> week of the quarter.

**NOTE:** IT department policy states that a student has one quarter to **challenge** any **grade**. After that, grades cannot be challenged.

**Course Description**

This is the third course in the introductory programming sequence required for all students majoring in Information Technology. Topics include advanced interface concepts, traditional programming data structures, programming utilities and reusability, introductory project design and management concepts and other concepts as time permits. Emphasis is placed on the development of problem-solving skills. Large programming assignments are required.

**Course Objectives**

**General:** The purpose of this course is to provide students with an introduction to the advanced concepts and skills needed to support the programming requirements of up-stream courses in the IT curriculum. Specifically, this course is intended to encourage students to continue to develop their problem solving skills, to begin building a “logical toolkit” of algorithms and data structures, and to understand the

benefits of reusability. Students should also grasp the basics of program design and management.

Contribution to Measurable Program Outcome(s):

- Make effective oral presentations
- Program effectively within the student's specialty area
- Apply a development life cycle to a problem
- Design and develop a software prototype
- Participate effectively as a team member
- Be able to work effectively with users of an organization
- Practice user-centered design, development, and deployment

**Specific:** At the end of this course, a student should be able to implement moderately large programming projects either individually and/or in a team. Specifically, a student should:

- Demonstrate the ability to create graphical user interfaces based on a problem description.
- Demonstrate the creation and use of reusable objects
- Demonstrate the use of the built-in data structure classes within a program.
- Be able to implement data structures from scratch as well as use language-supplied utility classes
- Be able to create, examine, and use byte based files, showing knowledge of the way data is represented.
- Determine when it is appropriate to use threads and demonstrate how to create a threaded program.
- Demonstrate how to communicate between two machines using the network programming classes
- Be able to describe several standard programming design patterns and how they can be applied to problem analysis
- Be able to design and implement a fairly large project as part of a team.

**Prerequisites:**

4002-218 or 4002-221

**Role of course in curriculum for:**

IT: This is the third course in the introductory programming sequence required for all Information Technology students.

**Course required for graduation in:**

BS/IT: Yes

BS/ANSA: Yes

## Organization

**Lecture-By-Example (LBE):** The course format combines lecture and hands-on practice. Each class lecture begins with an introduction to a new topic, and is then followed by writing an example program. As I explain each concept and code the new material through an example on my computer which is projected to the class, you simultaneously write the program as well on your lab computer. A typical format would be asking you what to write before coding key new material step by step. I would ask a randomly chosen or volunteering student each time, and record the student's performance for bonus points toward in-class practical exam. This will not only create a highly interactive environment between the students and the instructor, but also encourage the students to prepare for class. You are expected to prepare for class from reading assignment, lecture notes, and practice exercise problem (see below for details) posted on MyCourses.

**Practice Exercises:** Practice Exercises will be part of most class sessions. The Exercises will be collected and graded. The Exercises will normally be completed in full during the class session. However sometimes, extensions may be given such that they will be due at the beginning of the next class session. Not handing in a Practice Exercise for any reason will result in a grade of zero for that Exercise. There are no make-ups for missing Practice Exercises.

The practice exercise and lecture notes will be available on MyCourses prior to each class to help students prepare for that class session.

**Bring a flash drive with you to every class to save your work.** Make sure your name and section # are clearly marked on your disk.

**Homework:** Programming assignments will be assigned as an integral part of this course. This work is to be done outside of the usual class hours. In general, these assignments will be more complicated and larger than the practice exercises. Late submission of homework assignments will be penalized as follows:

- up to 1 week late: you will receive **80%** of your actual grade
- more than 1 week late: no credit
- no homework assignments will be accepted after the last day of class
- If you want to have timely feedback on your HW, you need to turn it in on time; otherwise it becomes my lowest priority.

**In-Class Practical Exams:** There will be two in-class practical exams given during the quarter to assess the student's programming skills during live programming sessions in front of the course instructor(s). **Notes, textbooks and calculators will not be allowed during these times (unless special accommodations permit them).**

Missing the practicum will result in a grade of zero **unless the student contacts the instructor at least 24 hours in advance of that practicum. If student's reason is valid (documentation may be required), the student will be allowed to take a different make-up practicum at a later time.**

**Final Project:** A final project will be part of this course. This will be a group effort, with small groups performing the design, implementation and testing of the necessary code. Design documentation and a user's manual will be required deliverables, in addition to fully commented source code and test cases.

In addition to the actual coding, you will be expected to deliver a presentation to the class, where your group will discuss your design, implementation and testing processes. You will also cover lessons learned while doing this project. This presentation will be in place of a final exam. All group members are expected to participate in the presentation.

**MyCourses Conference:**

There will be a MyCourses conference set up for this course. We will use this for class material distribution and submission of assignments (dropbox). You will need your ITS account for this.

**Course Outline**

**1.1 Concepts of Graphical User Interfaces (GUI)**

- 1.1.1 Review of Event Handling
- 1.1.2 Designing Event Handlers
- 1.1.3 Keyboard and Mouse generated Events

**1.2 Data Structures and Algorithm Design**

- 1.2.1 Stacks/Queues/Linked Lists
- 1.2.2 Using Environment-Provided Utility Classes vs. Designing As Needed

**1.3 Data Representation and Manipulation**

- 1.3.1 Character representation
- 1.3.2 Bit-level AND/OR
- 1.3.3 Byte IO concepts

**1.4 Multithreading Concepts**

- 1.4.1 The *Thread* Class
- 1.4.2 Thread States/Priorities/Groups
- 1.4.3 Thread Synchronization

**1.5 Network Programming**

- 1.5.1 Client/Server programming using socket classes
- 1.5.2 Using URL classes to retrieve data from the network

**1.6 Program Design and Implementation Concepts**

- 1.6.1 Component Construction
- 1.6.2 Design Process Documentation
- 1.6.3 SW Development Process Basics
- 1.6.4 Using Packages and Interfaces
- 1.6.5 Reusability Concepts

### 1.6.5.1 Patterns in SW Design

#### Grading

The grading scale used along with the grading criteria is as follows:

Component	Weight
Practice Exercises	5%
Homework	20%
In-Class Practical #1	25%
In-Class Practical #2	25%
Final Project	20%
Final Presentation	5%

Range	Grade
$\geq 90.0\%$	A
$\geq 80.0\% \ \& \ < 90\%$	B
$\geq 70.0\% \ \& \ < 80.0\%$	C
$\geq 60.0\% \ \& \ < 70.0\%$	D
$< 60.0\%$	F

#### Proposed Schedule (Subject to change at any time)

W e e k	Day	Subject	Assigned Reading (Liang6)	Activities / Assignments Due
1	1	Course Introduction 218/221 Review	Review 218/221 notes	<i>HW#1 out</i>
	2	Inner Classes	Liang6 Pg 469	
2	1	More GUI	218/221 Notes & Liang6 Ch 12, 13, & 15	
	2	Threads I	Liang6 Ch 24	<i>HW#1 due, HW#2 out</i>
3	1	Threads II	Liang6 Ch 24	
	2	Byte IO	Liang6 Ch 18	
4	1	Exam Review		<i>HW#2 due</i>
	2	Practical Exam #1		
5	1	Network Programming	Liang6 Ch 25	<i>HW#3 out</i>
	2	Network Programming	Liang6 Ch 25	
6	1	Specialized IO	Liang6 Ch 18 pp 617-620	Team Project Assigned
	2	Data Structures & Algorithms	Liang6 Ch 20	<i>HW#3 due, HW#4 out</i> Team member formation
7	1	Collection Framework Exam Review	Liang6 Ch 22	
7	2	Practical Exam #2		

8	1	Unicode & Internationalization Project Planning/ design final project	Liang6 Ch 26	
8	2	Sorting/Searching/Comparators JAR	Liang6 Ch 23	<i>HW#4 due</i>
9	1	Testing Strategies/ Work on final project		
	2	Presentation Skills/ Work on final project		
10	1	Work on final project		
	2	Complete/Demo final project		Project Due
11		Final project Presentation		Final Project Presentation Materials (Power Point Slides)

**Cell Phones/Pagers:** Absolutely no cell phones/pagers are allowed in class--nor may a person leave the room to answer a cell phone in the hall. Please contact instructor for any anticipated emergency situations.

**Notices of Accommodation:**

If you have a "Notice of Accommodation", you must provide with a copy within a week of starting this course. If you provide me with the notice later in the course, it will not be retroactive. (In other words, an NOA is not a license to retake an exam or practical that you have done poorly on.)

**Academic Honesty Policy:** Please review the departmental policy on cheating as described at <http://www.it.rit.edu/dishonesty.php>

**Student Responsibilities:** Please review the general student responsibilities as outlined at [http://www.rit.edu/~301www/student\\_conduct/StudentR\\_R/index.html](http://www.rit.edu/~301www/student_conduct/StudentR_R/index.html)

**Finally...**

Any or all of the previous information is subject to change or modification during the quarter.
-------------------------------------------------------------------------------------------------

## **ACADEMIC DISHONESTY POLICY DEPARTMENT INFORMATION TECHNOLOGY**

The following statement is the Policy on Academic Dishonesty for the Department of Information Technology:

The Department of Information Technology does not condone any form of academic dishonesty. Any act of improperly representing another person's work as one's own (or allowing someone else to represent your work as their own) is construed as an act of academic dishonesty. These acts include, but are not limited to, plagiarism in any form or use of information and materials not authorized by the instructor during an examination or for any assignment.

If a faculty member judges a student to be guilty of any form of academic dishonesty, the student will receive a **FAILING GRADE FOR THE COURSE**. Academic dishonesty involving the abuse of RIT computing facilities may result in the pursuit of more severe action.

If the student believes the action by the instructor to be incorrect or the penalty too severe, the faculty member will arrange to meet jointly with the student and with the faculty member's immediate supervisor. If the matter cannot be resolved at this level, an appeal may be made to the GCCIS Academic Conduct Committee.

If the faculty member or the faculty member's immediate supervisor feels that the alleged misconduct warrants more severe action than failure in the course, the case may be referred to the GCCIS Academic Conduct Committee. The Academic Conduct Committee can recommend further action to the dean of the student's college including academic suspension or dismissal from the Institute.

The following definitions will be used to clarify and explain unacceptable conduct. This is not intended to be an exhaustive list of specific actions but a reasonable description to guide one's actions.

**CHEATING** includes knowingly using, buying, stealing, transporting or soliciting in whole or part the contents of an administered/unadministered test, test key, homework solution, paper, project, software project or computer program, or any other assignment. It also includes using, accessing, altering, or gaining entry to information held in a computer account or disk owned by another.

**COLLUSION** means the unauthorized collaboration with another person in preparing written work or computer work (including electronic media) offered for credit. Final work submitted by a student must be substantially the work of that student. Collaboration on an assignment is expressly forbidden unless it is explicitly designated as a group project. When there is any doubt, a student should consult the instructor (**NOT ANOTHER STUDENT**) as to whether some action is considered collusion.

Whenever there is any question as to whether a particular action is considered academic dishonesty, the instructor should be consulted.

The penalty for academic dishonesty in a course is an automatic "F" in that course.