R·I·T	**Rochester Institute of Technology** **Golisano College of Computing and Information Sciences** **Department of Information Sciences and Technology**

GCCIS-ISTE-121 Computational Problem Solving for Information Domain 2
Fall 2018 (2181) – Course Syllabus

REMINDER: The information presented in this syllabus is subject to expansion, change, or modification throughout the semester

Instructor: Peter Lutz Office: 70-2345 Office Hours: Tuesday 9:30am-noon Other times available by confirmed appointment. ☎ Phone: (585) 475-6162 ✉ Email: Peter.Lutz@rit.edu TA:	Course: GCCIS-ISTE-121 Section 01 Name: Computer Problem Solving for Information Domain 2 Room: GOL 70-3510 (sect 01) Time: MWF 08:00am-9:50am (sect 01) myCourses is used to receive materials and submit work https://mycourses.rit.edu/

Course Text and Materials

"Big Java: Early Objects" (6e edition) by Cay S. Horstmann, Wiley, 2015, ISBN 978-1-119-18718-9.

Course Description

This is the second course in the introductory programming sequence required for all students majoring in Information Technology. Topics include GUI interface development, file I/O, 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.

Important RIT Deadlines

August 27 - Monday – Classes begin
September 4 - Tuesday – Last day of add/drop a course
November 9 - Friday – Last day to withdraw with a grade of "W"
December 10 - Friday – Last class
December 12-14, 17-19 – Final exam period

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

NOTE: Any work not submitted by the final due date receives a grade of zero, unless arrangements are made previous to the initial due date.

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 IST 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 analysis, design and project management skills.

Contribution to Measurable Program Outcome(s):

- 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 and/or leader
- Practice user-centered design, development, and deployment
- Make effective oral presentations

Specific: At the end of this course, a student should be able to implement moderately large programming projects both individually and 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 language-supplied data structure classes within a program.
- Be able to create, read and write character-based files, showing knowledge of the way data is represented.
- Be able to create, read and write 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 multi-threaded program.
- Demonstrate how to communicate between two machines using the network programming classes.
- Be able to design and implement a fairly large project as part of a team.

Prerequisites:

ISTE-120, or equivalent

Should you not have the prerequisite courses or knowledge, you must see the instructor within the first two days of classes.

Role of course in curriculum for:

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

Course required for graduation in:

BS/CIT: Yes
BS/WMC: Yes

Expectations and Organization:

Attendance: Each class presents new information that builds on the previous day's work. Missing a class can put the student's success in jeopardy. Attendance is mandatory and will be recorded.

Labs: A certain amount of in-class work is required during the lab sessions. **These assignments will be graded and sometimes collected**. Normally these exercises will be completed fully during the class session. There are no make-ups for missing Lab's. Should you not be able to complete the lab during class, it must be in the dropbox by the beginning of the next class, and the paper copy needs to be given to the TA.

Bring a flash drive/memory stick with you to every class or have personal online storage to save your work. Recommend creating a file named "_OwnedBy.txt" on any device containing your full contact information; name, phone number(s), email, postal mailing address, reward(?).

Homework: Programming assignments will be assigned as an integral part of your learning for this course. This work is to be done outside of the usual class hours, and should be viewed as a gauge of how well you understand the topics. Sometimes you may seek outside help to complete the homework. 'Good help' is of the form where someone explains the topics to a point you could apply the concept to any problem. 'Bad help' is of the form where you see code and use it, or a variation of it. This 'bad help' causes you to think you understand the topic but during an exam you do not know how to apply the concepts. This 'bad help' is considered academic dishonesty.

These homework assignments are more complicated and larger than the lab exercises. Homeworks will be assigned and due as posted in MyCourses. Assignments may be due every week, while others may have up to a two week due date. A new assignment could be assigned on or before the previous one is due, or there may be a few days between a due date and the next assignment being assigned. You may ask questions about the homework in class, one-on-one or in the MyCourses discussions.

Late Homework policy: The date/time stamp in MyCourses determines the date/time of submission, regardless of what time your computer or time zone says. Late submission of homework assignments will be penalized 20% starting after the due date/time. After the late date/time has passed, the homework is worth zero points. There are no individual extensions of homework. No homework assignments will be accepted after the last day of class for this semester. These policies are to encourage you to practice and learn the materials when they are covered so you can fully learn the next concepts in the course.
Drop boxes close on the last class day. This may change, depending on availability of grading. You may also resubmit homework's for re-grade, but the later date submitted determines the maximum grade, which may include late deductions.
Homework assignments are posted in MyCourses. It is your responsibility to pick up the homework and complete it by the posted due date.

All homework must receive a good attempt, and full participation in the final project to receive an A or a B in this course. A partially functioning homework is considered a good attempt. Failure to provide an attempt at homework or project may result in a letter grade deduction from your final course grade.

In-Class Practical Exams: There will be three in-class practical exams given during the semester 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).**

"Final Exam": The final exam time will be used as a presentation of your Final Project. This will be given during finals week as scheduled by the registrar's office.

Occasionally students have conflicts between final exams scheduled at the same time. If you have conflicting final exams, according to institute policy, *"…by the last day of the 10th week of classes during fall or spring term or in the case of summer term or other sessions less than 16 weeks by the last day of 2/3 of the session, the student should submit a written request for rescheduling to the head of the home department, with a copy to the instructor being asked to provide the rescheduled examination. By the last day of the 12th week of classes the department head will, after consultation with the parties involved, notify the student of the date of the rescheduled examination…*"

(http://www.rit.edu/~620www/Manual/sectionD/D11.html)

Practical Exams:

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.

NOTE: Students are required to have at least a 75% average of the exams in order to pass the course.

Projects*:* A group project is part of this course. This will involve small groups working on the design, implementation and testing of the necessary code. Design and other documentation will be required deliverables, in addition to fully commented source code and test cases.

In addition to the actual coding and other project activities, you will be expected to deliver at least two presentations 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 the final exam day. All group members are expected to participate in the final presentation.

Peer reviews will determine your participation effort for the final project. Failure to fully participate may result in a letter grade deduction.

MyCourses: You are required to check your MyCourses account for course communications, and for submission of those assignments that must be submitted in soft-copy (electronic) form to a designated dropbox. Should you have problems with submitting through MyCourses or to the email addresses listed in MyCourses, contact the MyCourses help center 475-HELP, or email the help desk and CC: your instructor with the problem.

Recommend checking your MyCourses account, look for this Syllabus online. If you don’t know how to use MyCourses for seeing content, or submitting to the dropbox, please seek instruction with the myCourses help, or ask the instructor/TA. This is your responsibility to get the assignment information, and dropbox your work.

MyCourses dropbox for this course is the only means where assignments are accepted. No work is accepted through email to the instructor, TA, or graders.

Grading

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

Component	Weight
Homework	25%
Lab work	10%
In-Class Practical #1	10%
In-Class Practical #2	15%
In-Class Practical #3	15%
Final theory exam	5%
Project *	25%

Range	Grade
>= 90.0%	A
>= 80.0% & < 90.0%	B
>= 70.0% & < 80.0% Or, *not attempting **all*** Homework assignments	C
>= 60.0% & < 70.0%	D
< 60.0% Or < 75% over-all exam average	F

Project* - 20%	Weight
Design document 1	5%
Project Plan	10%
Mid Project review	10%
Design document 2	10%
Presentation	10%
Final Project	45%
Peer review	10%

*Not **fully** participating* in projects or attempting homework's can deduct one letter grade from your final grade.
* Required deliverables may change. Weights will be adjusted.

Course Outline

- **1.1** Concepts of Graphical User Interfaces (GUI)
  - **1.1.1** GUI components
  - **1.1.2** GUI layouts
  - **1.1.3** Designing Event Handlers
  - **1.1.4** Keyboard and Mouse generated Events
- **1.2** Exceptions
  - **1.2.1** Checked and unchecked exceptions
  - **1.2.2** Custom exceptions
- **1.3** File I/O
  - **1.3.1** Exception handling
  - **1.3.2** Byte, binary, stream I/O
- **1.4** Threading Concepts
  - **1.4.1** Thread States and Priorities
  - **1.4.2** Thread Synchronization
  - **1.4.3** Multi-threading
- **1.5** Network Programming
  - **1.5.1** Client/Server programming
  - **1.5.2** Internet programming
- **1.6** Data Representation
  - **1.6.1** Stacks/Queues/Linked Lists
  - **1.6.2** Data Structures and Algorithm Design
  - **1.6.3** Recursion, Generic programming
- **1.7** Program Design and Implementation Concepts
  - **1.7.1** Using Packages and Interfaces
  - **1.7.2** Reusability Concepts
- **1.8** Project Management
  - **1.8.1** Design Process
  - **1.8.2** Documentation
  - **1.8.3** Task definition, resource assignment and tracking
  - **1.8.4** SW development process Basics
  - **1.8.5** Test cases
  - **1.8.6** Project tracking
- **1.9** Advanced topics (as time allow)
  - **1.9.1** XML
  - **1.9.2** Unicode and Internationalization

Course Schedule (subject to changes)

ISTE 121 Schedule (2181)					
Month	**Date**	**Day**		**Topic**	**Other**
August	27	M	1	Day01 - Introduction	Ch. 18
	29	W	2	Day02 - GUI Layout managers	HW01 out
	31	F	3	Day03 – Lab 1: GUIs	
September	3	M	--		LABOR DAY – No Classes
	4	T	--		LAST OF DROP/ADD
	5	W	4	Day04 - Events	HW01 due, HW02 out
	7	F	5	Day05 – TextAreas, Menus, Inner Classes	
	10	M	6	Day06 - Lab02 – TextAreas, Menus, Event Handling	
	12	W	7	Day07 – Binary Stream IO	
	14	F	8	Day08 – Binary IO Exercise and Proj Mgmt	HW02 due, HW03 out
	17	M	9	Day09 - Lab03 – GUIs and Inner Classes	
	19	W	10	Day10 – Threads Intro	
	21	F	11	Day11 – Thread Control I	HW03 due, HW04 out
	24	M	12	Day12 – Lab04 – Threads and Progress Bars	
	26	W	13	Day13 - Synchronization	
	28	F	14	Day14 – Thread Control II & Deadlock	
October	1	M	15	Day15 - Lab05 – Threads and Binary IO	HW04 due, HW05 out
	3	W	--		CAREER FAIR – Class Canceled
	5	F	16	Day16 – Practical Exercise – ByteIO, Threads	
	8	M	--		OCTOBER BREAK – No Classes
	10	W	17	Day17 – Practical Exercise – Threads50	
	12	F	18	Day18 – Practical 1 – Threads and Binary IO	
	15	M	19	Day19 – Networking, Client and Server	
	17	W	20	Day20 – Multithreaded TCP Server and UDP	HW05 due, HW06 out

	19	F	21	Day21 – Lab06 / HW07 – Networking Client/Server	
	22	M	22	Day22 – Lab06 / HW07 Continued	
	24	W	--	UTILITY DAY	For use as instructor sees fit.
	26	F	23	Day23 – Lab07 / HW08 – Object IO over Sockets	HW06 due
	29	M	24	Day24 – Lab07/HW08 Continued & Generics	Final Project Discussion
	31	W	25	Day25 – Sorting and Searching	
	2	F	26	Day26 – Sorting More Complex Data	
November	5	M	27	Day27 – Lab08 - Sorting	Lab06 / HW07 due
	7	W	28	Day28 – Project Day – Communication, Protocol Design	
	9	F	29	Day29 - Recursion	LAST DAY TOI WITHDRAW WITH A "W"
	12	M	30	Day30 – Lab09 – Recursion and Directory Tree Walking	Lab07 / HW08 diue
	14	W	31	Day31 – Practical 2	
	16	F	32	Day32- Project Day – Design Review	Teams present design overview At class / interaction / UML level of detail
	19	M	33	Day33 – Stacks, Queues and Lists	
	21	W	--		THANKSGIVING BREAK – no class
	23	F	--		THANKSGIVING BREAK – no class
	26	M	34	Day34 – Sets and Maps	
	28	W	35	Day35 - Lab10 – Data Structures	
	30	F	36	Day36 – Jars and Packages	
	3	M	37	Day37 – Project Day – Code Review	Teams present code for one or more major components of their project.

December	5	W 38	Day38 – Project Day – Interoperability Testing	
	7	F 39	Day39 – Practical 3	
	10	M 40	Day40- Project Day – Interoperability Testing	LAST DAY OF CLASS
	11	T	READING DAY	
	12-14 17-19		EXAMS – Demo Day	Final Project Due – this is at the scheduled final exam time.

Additional Policies:

Cell Phones/Pagers: Absolutely **NO** ringing cell phones/pagers or other devices are allowed in class nor may a person leave the room to answer a call in the hall. Please contact instructor before class for any anticipated emergency situations.

Notices of Accommodation: If you have a “Notice of Accommodation”, you must provide me 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.

Homework Grading: Programs that do not compile, with the Java version used in the RIT labs, can expect to receive a grade of zero. This also includes forgetting to send Java files or classes required for your application to run by itself. Don’t expect the instructor to fix errors or create classes for your applications. Recommend deleting your *.class files, compile everything and run multiple tests before sending to the dropbox. Resubmitting work to include what was originally missing after the due date will have late penalty applied as applicable.

Late Policy: If you are having problems with an assignment or an emergency that may make you late in submitting your work, contact the instructor before the due date. Excuses made after the fact will not be honored. This applies to Lab Exercises, Homeworks, Exams, and Final exam.

Extra Credit: The policy on extra credit is simple: I do not offer extra credit assignments for any reason. Individual extra credit work does not exist; do not ask for any. Extra points on an assignment may be part of the assignment to gain extra points for trying to apply more difficult concepts to your work. Take advantage of these opportunities.

Final Exam:

The final exam date for this course is set by the Registrar’s Office at the beginning of the semester. Do not make travel plans without checking the exam schedule. You being present for the final presentation of your work is required. No accommodation will be given to accommodate your plans.

Final Project Presentations: The presentation of your team’s final project is in week 16. This is in a “Trade Show” format, where each team sets up their project from presentation and demo. Since the final exam time is only 2 hours, this allows about 10-15 minutes per team. So you must be efficient in your presentation. **Do not make travel arrangements prior to this** time no accommodations will be made. Part of your grade depends upon your full participation in the final trade show of your work.

Java Version: We use the current version of J2SE as of the beginning of the semester in the labs. You should be using a similar version for your assignments. See “Homework Grading” above to see how code that does not compile is graded.

Tutors: The IST department has several Java tutors. They are there to help with specific programming problems. They will not give you the answer to homework, but can help you work through debugging, logic problems, etc. If you do not understand basic course concepts, you should come to office hours or request alternate office hours. Tutors are not there to teach the course, I can help you with your understanding of topics better in a one-on-one environment when needed.

Contact Information:
Professor to Student Communication: Any updates to assignments and any emails that I need to send to individual students will be done through MyCourses. What this means is that you should check your email and the MyCourses conference for this course periodically. If you forward your email to another account, you are responsible for making sure email forwarding continues to work throughout the semester.

Student to Professor Communication: I generally have email running whenever I am logged in, so you should get a reply to any email you send me within a day. Except for Fridays and weekends which may be a bit longer as I am generally out of Internet service capability on some weekends.
To separate your email from spam, start your **email subjects with "121-0x…"** so I know what course and section you are taking and I don't ignore it as spam.

Academic Honesty Policy: Please review the institute's policy on cheating as described at https://www.rit.edu/academicaffairs/policiesmanual/d080.

Should you encounter cheating in your team and you do not want to be associated with the work, first inform your team you are not going to be part of this, and request from the instructor that you be disassociated from the team and allowed to complete the project or that part of the work yourself, or with another team.

Note that if you get accused of cheating, other faculty members have already checked the evidence to verify it will withstand an appeal.

You may not use work that you, your team, or someone else wrote in a previous class. This is a frequently overlooked clause in the Academic Honesty Policy.

Finally…

I do not tolerate any form of cheating in this course. If you are caught, you will receive the full departmental penalty: an F in the course, immediate removal from the class, and a letter detailing the incident will be placed in your folder.

R·I·T

Rochester Institute of Technology
Golisano College of Computing and Information Sciences
Department of Information Sciences and Technology

ACADEMIC DISHONESTY POLICY
DEPARTMENT OF INFORMATION TECHNOLOGY

As a university, RIT is committed to the pursuit of knowledge and the free exchange of ideas. In such an intellectual climate it is fundamentally imperative that all members of this academic community behave in the highest ethical fashion possible in the manner by which they produce, share, and exchange this information. In the case of students [1], Academic Honesty demands that at all times student work be the work of that individual student [2], and that any information which a student uses in a work submitted for evaluation be properly documented. Any violation of these basic standards constitutes a breach of Academic Honesty and hence becomes Academic Dishonesty.

ACADEMIC DISHONESTY

Academic Dishonesty falls into three basic areas: cheating, duplicate submission and plagiarism.

1. Cheating - Cheating is any form of fraudulent or deceptive academic act, including falsifying of data, possessing, providing, or using unapproved materials, sources, or tools for a work submitted for faculty evaluation.
2. Duplicate Submission - Duplicate submission is the submitting of the same or similar work for credit in more than one course without prior approval of the instructors for those same courses.
3. Plagiarism - Plagiarism is the representation of others' ideas as one's own without giving proper credit to the original author or authors. Plagiarism occurs when a student copies direct phrases from a text (e.g., books, journals, internet) and does not provide quotation marks, or paraphrases or summarizes those ideas without giving credit to the author or authors. In all cases, if such information is not properly and accurately documented with appropriate credit given, then the student is guilty of plagiarism.

Consequences of Academic Dishonesty

Any act of Academic Dishonesty will incur the following possible consequences. After notifying and presenting the student with evidence of such misconduct, the instructor has the full prerogative to assign an "F" for the offense, or to assign an "F" for the entire course. The instructor will inform and, if possible, meet with the student concerning the decision reached on the "F" for the offense, or the "F" for the entire course. A student may be brought before the Academic Conduct Committee of the College in which the alleged offense occurred, and may face academic suspension or dismissal from the Institute. (See “D08.0 Student Academic Integrity Policy" https://www.rit.edu/academicaffairs/policiesmanual/d080 and "D18.0 Interim Policy - Student Conduct Process" https://www.rit.edu/academicaffairs/policiesmanual/d180)

Approved September 1977
Revised May 18, 2002

NOTES

[1] The policy for faculty ethical behavior is contained in C2.0 of the Institute Policies and Procedures Manual.

[2] On occasion student work may be in the form of a group project assigned and sanctioned by an instructor or group of instructors.

> The penalty for academic dishonesty in a course is an automatic “F” in that course.