

RIT - GCCIS COURSE SYLLABUS
ISTE.422.01 APPLICATION DEVELOPMENT PRACTICES
FALL 2019 (TERM 2191) DRAFT OF AUGUST 23, 2019

DETAILS

Important note: The information presented in this syllabus is subject to expansion, contraction, change, or stasis during the semester. In case of conflict between versions, the most recent copy on myCourses takes precedence.

Course Number. NA

Prerequisites.

- ISTE121 or
- ISTE200 or
- equivalent programming course(s), i.e., a second semester course in application programming using Java

Time. NA

Place. NA

Dates. NA

Final Exam. NA

Instructor. Mick McQuaid

Email. mjmics@rit.edu

Office. 2675 Golisano Hall

Office Hours. NA

DESCRIPTION

In this course, students will gain experience with the processes, practices, and tools professional developers use to deliver robust and maintainable applications. Students will apply these practices and tools to build smaller-scale production-quality applications and systems. Topics include development

life cycles, version control, test bed development and use, build utilities, error handling, deployment tools, and documentation.

MATERIALS

Each student will need at least a 32GB USB3 drive to bring to class every day. The student may substitute a larger drive or a laptop running VMWare, which is available free of charge to RIT students. This drive is needed to hold a virtual machine image used in class. The image is too large to quickly download each class period, necessitating the drive.

The course materials include readings on myCourses and the open Web and open source software required to complete exercises and the project. Students with experience using equivalent proprietary or other open source software may use it at the discretion of the instructor and with the understanding that the student bears the burden to make optional software work. The failure of optional software to perform as the student expects may lead to a failing grade for the student. The student may not simply blame the software nor may the student expect the instructor to troubleshoot the software.

Textbook. There is no textbook for the course. Required readings will be available on the open Web or in myCourses. Additional readings will be recommended but not required.

Technology. Five kinds of software will be used in this course. The first is a class of software packages designed to assist in application development. The second is a scripting language. The third is an application language. The fourth is a set of common shell utilities. Fifth is a text editor.

The student will be introduced to software packages designed to assist in application development. While most are language agnostic, they will be demonstrated using Java as

the application language.

The student will need to program in one scripting language, JavaScript. Other scripting languages may be substituted based on student expertise and at the discretion of the instructor but JavaScript will be the only scripting language demonstrated.

The student will need to program in one application language such as Java. No application language will be taught in this course and other full-featured languages may be substituted for some but not all application programming work in the course.

The student will need to manipulate certain shell utilities, including GNU coreutils. These will be demonstrated in class.

The student will need to use a text editor. Any text editor will be acceptable for most assignments and the project, but the open source text editor Vim will be demonstrated and used for at least one exercise and no text editor without comparable features may be substituted for that exercise.

SCHEDULE

The estimated course schedule follows. All dates, lecture topics, and assignments are subject to reasonable change at the discretion of your instructor. Any changes will be announced in class. Numbers refer to weeks of the semester.

1. Introduction / Baseline Setting
2. Development Methodologies
3. Diagramming Development
4. Version Control
5. Build Utilities
6. Test Bed Development and Use
7. Error Handling; Logging

8. Bug Tracking; Profiling
9. Generic Code; Data-driven Code
10. Reverse Engineering
11. Efficient Code
12. Application Deployment
13. Help Systems
14. Packages, Frameworks, Namespaces, JARs, DLLs
15. Documentation

GRADING

The grading scale used along with the grade components follow. Numbers refer to weeks of the semester.

- A $\geq 90.0\%$
 - B $\geq 80.0\%$ & $< 90\%$
 - C $\geq 70.0\%$ & $< 80.0\%$
 - D $\geq 60.0\%$ & $< 70.0\%$
 - F $< 60.0\%$
1. Exercise 1, 05% (chaos report)
 2. Exercise 2, 05% (improvised etl)
 3. Exercise 3, 05% (version control)
 4. Exercise 4, 05% (make)
 5. Exercise 5, 05% (build)
 6. Exam 1, 05%
 7. Milestone 1, 05% (test plan)
 8. Exercise 6, 05% (logging)
 9. Milestone 2, 05% (sdhc)
 10. Exam 2, 10%
 11. Exercise 7, 05% (test fixture)
 12. Milestone 3, 05% (deployment strategy)
 13. Milestone 4, 10% (help system)
 14. Exercise 8, 05% (profiling)

15. Milestone 5, 10% (refactored abstracted code)
16. Exam 3, 10% (finals week)

Grading Details. Adding the points from the above list shows that the course grade is composed of

- 35% project milestones
- 25% exams
- 40% in-class exercises

Project Milestones. Project milestone grades are shared by all members of a group. Peer evaluations may result in different scores based on the instructor's judgment.

Milestone 1. Test Plan

Milestone 2. SDLC

Milestone 3. Deployment Strategy

Milestone 4. Help System

Milestone 5. Refactored Abstracted Code

Exams. Students will face three unequally weighted exams.

Exam 1. Development methodologies through build utilities

Exam 2. Testing through reverse engineering

Exam 3. Efficient code through documentation

Exercises. Students will work on exercises in class. Some may be completed in groups while others need to be completed individually. The instructor needs to be able to observe progress on the exercises to support mastery of the skills involved so, while exercises may be finished outside class, major work on exercises needs to be completed in class in view of the instructor.

Exercise 1. Chaos Report

Exercise 2. Improvised ETL

Exercise 3. Version Control

Exercise 4. Make

Exercise 5. Build Utilities

Exercise 6. Logging

Exercise 7. Test fixture

Exercise 8. Profiling

POLICIES

The following are brief statements of policy that are, in many places, expanded at the URLs provided. You are bound by these policies and any protest that you did not read the extended versions at the provided links will not be heeded. Your familiarity with the following policies, dates, and parameters will be assumed in this course.

Last day of 7-day add/drop period. Tue 23 Jan 2018

Last day to withdraw with W. Fri 6 Apr 2018

myCourses. All project assignments, lecture notes, and other distributable course materials will be available via myCourses. Except where otherwise indicated, all student project assignments will be submitted via myCourses dropboxes.

Grade Challenges. School of Information policy states that a student has one semester to challenge any grade. After that, grades cannot be challenged.

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

Extra Credit. No extra credit is available in this course.

Accommodations. If you have a “Notice of Accommodation”, you must provide your instructor with a copy of it within 1 week of starting this course. You must follow all the rules of the relevant office.

Academic Dishonesty. The policy on dishonesty is simple: Anyone caught cheating receives an “F” for the course, is removed from the course, and a letter detailing the incident is placed into his or her folder. By the time an instructor accuses a student of cheating the evidence has likely been reviewed by other faculty members and will likely withstand an appeal. Please review the institute policy at http://www.rit.edu/studentaffairs/studentconduct/rr_academicdishonesty.php

Acceptable Use. We are bound by the following Acceptable Computer Use policy at <http://www.rit.edu/academicaffairs/policiesmanual/sectionC/C82.html>

Student Responsibilities. Please review the general student responsibilities as outlined at <http://www.rit.edu/~301www/rr.php3>

Policy on Reporting Incidents of Discrimination and Harassment. RIT is committed to providing a safe learning environment, free of harassment and discrimination as articulated in our university policies located on our governance website. RIT's policies *require faculty to share information* about incidents of gender based discrimination and harassment with RIT's Title IX coordinator or deputy coordinators, regardless whether the incidents are stated to them in person or shared by students as part of their coursework. RIT Governance website: <https://www.rit.edu/academicaffairs/policiesmanual/policies/governance>

If you have a concern related to gender-based discrimination and/or harassment and prefer to have a *confidential* discussion, assistance is available from one of RIT's confidential resources on campus:

1. The Center for Women & Gender: Campus Center Room 1760; 585-475-7464; CARES (available 24 hours/7 days a week) Call or text 585-295-3533.
2. RIT Student Health Center – August Health Center/1st floor; 585-475-2255.
3. RIT Counseling Center - August Health Center /2nd floor - 2100; 585-475-2261.
4. The Ombuds Office – Student Auxiliary Union/Room III4; 585-475-7200 or 585-475-2876.
5. The Center for Religious Life – Schmitt Interfaith Center / Rm 1400; 585-475-2137.
6. NTID Counseling & Academic Advising Services – 2nd Floor Lyndon B. Johnson; 585-475-6468 (v), 585-286-4070 (vp).