CICS 110

Foundations of Programming

General Information

Semester:	Fall 2022
Start Date:	September 6, 2022
End Date:	December 12, 2022
Credits:	4
Lectures:	**01-LEC(48638)**: MoWe 4:00PM-5:15PM, Lederle Grad Res Ctr rm A301, Cole Reilly
	02-LEC(56747): MoWe 4:00PM-5:15PM, Morrill Sci. Ctr.1 rm N326, Ghazaleh Parvini
	03-LEC(56749): MoWe 4:00PM-5:15PM, Morrill Sci. Ctr.1 rm N326, Ghazaleh Parvini
	04-LEC(56767): TuTh 11:30AM-12:45PM, Morrill Sci. Ctr.1 rm N326, Jakob Falus
	05-LEC(56770): TuTh 4:00PM-5:15PM, Flint Laboratory room 105, Cole Reilly
	06-LEC(56792): MoWe 2:30PM-3:45PM, Flint Lab room 201, Simon Andrews
Labs:	**01LL-LAB(48639)**: Fr 10:10AM-11:00AM, Integ. Learning Center N101, Staff
	02LL-LAB(56748): Fr 10:10AM-11:00AM, Hasbrouck Laboratory room 134, Staff
	03LL-LAB(56750): Fr 10:10AM-11:00AM, Hasbrouck Laboratory room 134, Staff
	04LL-LAB(56768): Fr 10:10AM-11:00AM, Flint Laboratory room 105, Staff
	05LL-LAB(56771): Fr 2:30PM-3:20PM, Flint Laboratory room 105, Staff
	06LL-LAB(56793): Fr 2:30PM-3:20PM, Flint Lab room 201, Staff
Prerequisites:	R1 (or a score of 20 or higher on the math placement test Part A), or one of the following courses: MATH 101&102 or MATH 104 or MATH 127 or MATH 128 or MATH 131 or MATH 132.
Instructors:	Ghazaleh Parvini, Cole Reilly, Jakob Falus, Simon Andrews
Staff	TBD

Description

An introduction to computer programming and problem solving using computers. This course teaches you how real-world problems can be solved computationally using programming constructs and data abstractions of a modern programming language. Concepts and techniques covered include variables, expressions, data types, objects, branching, iteration, functions, classes, and methods. We will also cover how to translate problems into a sequence of instructions, investigate the fundamental operation of a computational system and trace program execution and memory, and learn how to test and debug programs. No previous programming experience required.

Statement of Inclusivity

The staff for this course support the UMass commitment to diversity, and welcome individuals regardless of age, background, citizenship, disability, sex, education, ethnicity, family status, gender, gender identity, geographical origin, language, military experience, political views, race, religion, sexual orientation, socioeconomic status, and work experience. In this course, each voice in the classroom has something of value to contribute. Please take care to respect the different experiences, beliefs and values expressed by students and staff involved in this course.

Course Goals

The objective of this course is to introduce the fundamentals of computing and programming using a general-purpose programming language from a modern perspective. This includes understanding the operation of a machine from a programming language perspective and what it means to execute a whole program as well as its individual parts, how to solve problems using constructs that a programming language provides such as variables, data types, objects, branching, iteration, functions, and classes, and how to write programs that receive data from various sources, process that data, and produce output in various forms.

Learning Outcomes

At the completion of this course you will be able to:

- Read and write programs to solve non-trivial programs using the Python programming language.
- Describe fundamental units of computation and program structure.
- Translate real-world problems into computational solutions.
- Solve problems using a general-purpose programming language and the tools it provides such as variables, data types, objects, branching, iteration, functions/methods, and classes.
- Design and implement whole programs and functions to solve computational problems using top-down and bottom-up techniques.
- Describe application programming interfaces (API) and use APIs available from existing objects and libraries to solve problems.
- Translate data from and into various formats that are in computer memory, a graphical interface, a file, from a remote API on the web, or other data resources.

- Use console-based or graphical interfaces to learn about input/output to move data into and out of a program.
- Use modules and classes to organize data and functions.
- Explain the notion of a machine and how it relates to the execution of a general-purpose programming language.
- Explain how programs and their data are represented in a computer and build mental models and use diagrams of program and function execution and data stored in memory.
- Use basic debugging techniques such as “print debugging” and assert statements to determine the cause of logical programming errors and show the correctness of a program and its implementation.
- Describe programs using proper documentation techniques to communicate implementation details at various levels of granularity.

Course Format

Lecture:	Lecture will begin with a brief review of what was covered in the previous lecture followed by a presentation of new material. This presentation may include slides as well as code demonstrations that you will have access to as part of the course material. This will often be followed by an exercise that will help solidify your understanding of the material.
Lab:	Lab will have an associated assignment that you are required to complete. You are expected to complete these activities during the lab working with others taking the course. Course staff will be available to answer questions and help guide you through the assignment.

How to Succeed

Your success in this class is important to us. We all learn differently and bring different strengths and needs to the class. If there are aspects of the course that prevent you from learning or make you feel excluded, please let us know as soon as possible. Together we’ll develop strategies to meet both your needs and the requirements of the course. There are also a range of resources on campus, including:

- Academic Calendar
- Learning Resource Center
- Center for Counseling and Psychological Health (CCPH)
- English as a Second Language (ESL) Program

Course Material

Textbook

The following textbook is required:

- *Programming in Python 3*, an interactive textbook from zyBooks designed specifically for this course.

To activate your zyBook subscription:

- Go to: https://learn.zybooks.com
- Create an Account. Make sure to sign up with your **@umass.edu** email address.
- When prompted for the zyBook code, enter: XXXXXXXXXXXXXXXXXXXX
- You can purchase the Zybook online with a credit card. A subscription is $$$. You may begin subscribing on DDDDDDDD, and the cutoff to subscribe is DDDDDDDD. Subscriptions will last until DDDDDDDD.

Here are some additional textbook recommendations freely available online. You may consider looking at these as supplemental material:

- Automate the Boring Stuff with Python, Al Sweigert, https://automatetheboringstuff.com
- A Byte of Python, https://www.gitbook.com/book/swaroopch/byte-of-python/details
- Dive into Python, Mark Pilgrim, http://getpython3.com/diveintopython3
- Learn python the hard way, Zed Shaw, http://learnpythonthehardway.org/book
- Python Practice Book, Anad Chitpothu, http://anandology.com/python-practice-book

Laptop Computer

It is highly recommended that you have a laptop computer. We will be writing code both in and out of class, so a portable computer capable of installing software (not a Chromebook) is valuable for this class. Most in-class programming activities will be group-based, so if you do not own a laptop, you can easily work with another student in class.

Software Platforms and Tools

Moodle

We will use the Moodle Learning Management System (LMS) as the primary hub for course content. You will be able to access readings, lecture material, assignments, and any other important material pertaining to this course. We may use Moodle to submit some assignments. You will be able to access your latest grades and comments for assignments using the Moodle Gradebook.

zyBooks

The textbook for this course is available from zyBooks. Not only is zyBooks the book for the course, but it also includes visual and interactive content that increases your understanding of the material. It provides participatory content, challenge exercises, and a Python programming environment built right into the book!

Gradescope

We use Gradescope for automatically grading programming projects. Gradescope allows us to provide fast and accurate feedback on your work. Before the deadline you can submit as many times as you need, so submit early and often to ensure you have something in before the deadline. Become familiar with Gradescope and verify that your submission has been properly uploaded before the deadline. Use OneDrive, DropBox, Google Drive, or some other backup software to ensure that your work is not lost in the event of a computer failure. The Gradescope autograder will provide you with some limited feedback on your submissions: does it compile,

does it pass automated tests, what your score is, etc. The autograder does not provide detailed feedback. We will help you get familiar with Gradescope as the course progresses.

VSCode

We use the VSCode Development Environment for developing, debugging, and testing programming projects. This is free software and you will be given installation instructions and training in its use. There are many excellent Python integrated development environments (IDE), editors, and tools that exist, however, we recommend VSCode as it is easy to learn, and we will be using it in class for code demonstrations. You can read more about VSCode here and we encourage you to try out the "Getting Started with Python in VSCode" as preparation for the class.

Python

We will be using the Python programming language in this course. It is an excellent language to learn about programming and computational thinking. It is also used extensively in the software engineering, data science domain, and many additional areas of computing. You will be required to download and install Python on your own computer. You will be using Python and VSCode to complete many programming tasks as part of your studies in this class. The current Python version used in this course is 3.10.6.

Communication

Email

Email should not be used. **Please post privately to Instructors on Piazza**.

In the unlikely event that you are unable to post to Piazza, please send an email to the instructor teaching your course section.

Piazza

We will be using Piazza for all other communication. This online discussion forum should be your first choice for asking questions. You should check the discussion forum before asking your question to see if the same question has already been posted. We will not answer questions that have already been answered in the discussion forum. Think before you post. We expect you to do a reasonable amount of thinking to try to solve your problems before posting for help. Make sure you are articulate and clear with your post (i.e., think before you post). You should post questions related to assignments early rather than wait until the last minute. Questions that are posted very near an assignment deadline may not be answered. Course staff are expected to answer questions Monday through Friday. Do not expect prompt answers on Saturday, Sunday, and scheduled holidays and breaks.

Please post with respect and kindness. Posts that are disrespectful, crude, inappropriate, or mean will not be tolerated and will be reported and result in your immediate removal from the course and a failure for the course.

Attendance

We expect you to attend lectures and labs on a regular basis. If you are absent or miss deadlines for health reasons or other extenuating circumstances, you will be able to view the lectures online as they will all be recorded on echo360. If you do miss class or lab and there is an assignment to complete you must notify us as soon as possible and, if you seek excusal from an assignment or require an extension, to provide written documentation.

Assessment and Grading

The final grade for this course is broken down into the following categories and weights:

- 10% Lab Assignments
- 20% Homework Assignments
- 40% Programming Projects
- 20% In-Class Exams
- 10% Final Exam

The numerical cutoff for final course letter grade assignment will be made after all grading is completed. As a rough guide, expect to require at least a 93 to get an A, a 90 to get an A-, an 87 to get a B+, an 83 to get a B, an 80 to get a B-, etc.

Individual grade items are not curved, so you should not get stressed about means, standard deviations, etc. related to scores you receive. What matters is your weighted average; we do not give favorable (or unfair) treatment by raising or lowering individual students' letter grades.

There are no opportunities for extra credit in this course; please do not ask.

You are responsible for monitoring your grades. Grades will be available through Moodle and you should check them regularly and review any provided feedback. If you encounter any issues with your grades, you will have one week past the first posting of a particular assignment's grade to Moodle to contact the course staff so that we can investigate.

Lab Assignments

Most labs will be accompanied by an assignment that you will complete totally or partially during your assigned lab section. You will complete 1 to 3 programming exercises during the lab. If you are unable to complete the assignment during that time you will be responsible for completing the work on your own time before the assigned due date. You may work collaboratively during the lab section, but submit your work individually. Labs will be assigned at the start of the lab in Moodle and completed and submitted in the zyBook. They will be automatically graded by the zyLab autograding system. During the lab section you will have the opportunity to ask the lab staff questions. The lab staff will provide guidance so that you may complete the lab successfully. Lab assignments are low-stakes exercises that are designed to allow you to practice your understanding of the material covered in the book and during lecture in the presence of the course staff in a collaborative setting. You will need to bring a laptop if you have one or work with another student. If you do not have a laptop, please notify the course staff so we can make arrangements for you to work with another student.

Homework Assignments

There will be a number of homework assignments that you will complete individually by the assigned due date. Each homework assignment will have you complete a programming challenge by analyzing the problem, coding the solution, answering some questions about the problem, and providing a reflection on your experience before you hand in your solution. Homework assignments will be completed using Zylab and/or VSCode and your completed assignment will be submitted to Zylabs or Gradescope. Homework assignments are a low-stakes exercise that are designed for you to practice problem solving with code independently.

As you complete a homework assignment you should take note of your abilities with respect to the material. Although getting the correct code is an important part of programming, that is not the only criteria you are evaluated on and it is not necessarily the purpose of a homework assignment. You should use the homework as an instrument to determine how well you are understanding the material.

Programming Projects

There will be approximately 4 programming projects. Programming projects are high-stakes programming assignments that will assess you on what you have learned. A programming project is a more in-depth assignment where you will apply the programming concepts you have learned to solving a problem. You will need to demonstrate your understanding of the concepts and the programming skills you have learned from the zyBook, lecture, lab, and homework. Programming projects are completed independently using Python/VSCode and are automatically graded by Gradescope. Gradescope will run several tests on your program to evaluate your solution. You will be allowed to submit a programming project as many times as you would like before the deadline. Programming projects measure your mastery of the learning outcomes.

Exams

There will be 3 in-class exams. Your best two scores will count towards your final grade. Exams are high-stakes assignments that will assess your ability to recall, understand, and apply the course material covered in the zyBook, lecture, lab, homework, and programming projects. Each exam focuses on specific lectures and associated material and assignments. Due to the nature of the material, all exams are cumulative.

Exams will be taken during a scheduled class session. You will have the entire class period to complete the exam and submit it. Questions consist of multiple choice, true/false, fill in the blank, matching, reading/writing code and other similar forms.

To prepare for an exam you should review all the readings, lecture material, code, and exercises/assignments covered by the exam. It is important that you plan ahead and give yourself plenty of time to review the material. Cramming for an exam or pulling an all nighter can lead to anxiety and exhaustion and often result in lower performance. So, be kind to yourself and begin preparations by studying the material often and early.

Final Exam

The final exam will be given during the final exam period. The final exam will be cumulative and assess your knowledge of the material covered during the entire semester. You will have the entire exam period to complete the exam and submit it. Questions consist of multiple choice, true/false, fill in the blank, matching, reading/writing code and other similar forms.

To prepare for the final exam you should review all the readings, lecture material, code, and exercises/assignments covered in the course. It is important that you plan ahead and give yourself plenty of time to review the material. Cramming for an exam or pulling an all nighter can lead to anxiety and exhaustion and often result in lower performance. So, be kind to yourself and begin preparations by studying the material often and early.

Make sure you get plenty of rest and eat well before you arrive at the final exam!

Late Submissions

It is your responsibility for maintaining your own schedule and being prompt with your submissions. We expect that you become familiar with the course submission software and verify that your submission has been properly uploaded. We will not grant late submissions due to lack of checking on this. We assume:

- You are an adult and have the ability to check and verify your work has been received properly.
- You are capable of using OneDrive, DropBox, Google Drive, or some other backup software to ensure that your work is not lost in the event of a computer failure.
- You have a back-up plan in place in the event that your computer fails or your internet connection is unavailable. Make sure you have a plan B and C if your computer crashes or your internet is unavailable. This is your responsibility.

To ensure that you submit assignments on time you should begin them early and not wait until the last minute to submit. For some assignments you may be able to submit multiple times so submit early and often to ensure you have something in before the deadline.

Late Submission Requests

Requests will be reviewed and responded to in a timely fashion. If you are requesting an extension to a deadline and the request is granted, arrangements will be made for you to complete the work. It is your responsibility to complete that work by the extension deadline.

Although you are allowed to make requests for extensions, we are not obligated to grant the extension request. If we find that you are consistently requesting extensions, we will schedule a meeting with you to discuss why you are unable to complete assignments on time and determine the best path forward.

Please send late submission requests to Instructors on Piazza.

Incompletes

Typically, a course is completed after the last class, final exam, and/or final project or assignment. In rare cases, extenuating circumstances may prevent a student from completing a

course by that time. As part of the University Regulations, we may issue an Incomplete (INC) for a course, rather than a course grade, if a student submits a request to the instructor(s). The criteria for granting an INC request are determined by the course instructors. The following is an excerpt from Section VI D in the Academic Regulations:

> *"Students who are unable to complete course requirements within the allotted time because of severe medical or personal problems may request a grade of Incomplete from the instructor of the course. Normally, incomplete grades are warranted only if a student is passing the course at the time of the request and if the course requirements can be completed by the end of the following semester. Instructors who turn in a grade of "INC" are required to leave a written record of the following information with the departmental office of the academic department under which the course is offered: (1) the percentage of work completed, (2) the grade earned by the student on the completed work, (3) a description of the work that remains to be completed, (4) a description of the method by which the student is to complete the unfinished work, and (5) the date by which the work is to be completed. In the case of an independent study where the entire grade is determined by one paper or project, the instructor should leave with the department information pertaining to the paper or project, which will complete the course. To avoid subsequent misunderstanding, it is recommended that the student also be provided with a copy of this information."*

Criteria

The incomplete criterion for this course requires that you have:

- At least 60% of the course must be completed with a passing grade.
- A valid reason for requesting an INC that relates to a severe medical or personal problem.

Requests

Towards the end of the semester a notification will be posted about incomplete requests. You will follow the instructions provided to submit an incomplete request. After we review the request, we will make one of the following determinations:

1. We **approve** the request upon which you will be notified by email and a separate incomplete agreement document will be sent to you to read through and sign no more than 48 hours after receiving the incomplete agreement document. This document will include what remains to be completed for the course and a deadline. After you sign and return this document, we will open extensions for the missing work. After the course has ended, we do not provide any additional help or support regarding the specifics of the course material. You are expected to complete the work using the material and online platforms that were available to you when the course was active.
2. We **deny** the request and submit a grade based on your performance at the end of the course.

Course Support

Office Hours

Office hours are times when we provide real-time access to the instructor, TAs, and UCAs. You do not need an appointment to attend office hours, attendance is optional, and all questions you have about the course are welcome. These sessions will be held at different times during the week. Office hours will be posted on the course website. Office hours will be held both in person and on Zoom.

Accommodations

The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented physical, psychological, or learning disability on file with Disability Services (DS), you may be eligible for reasonable academic accommodations to help you succeed in this course. If you have a documented disability that requires an accommodation, please notify your instructor as soon as possible so that we may make appropriate arrangements. For further information, please visit Disability Services (https://www.umass.edu/disability).

Title IX

If you have been the victim of sexual violence, gender discrimination, or sexual harassment, the university can provide you with a variety of support resources and accommodations. UMass is committed to providing these resources with minimal impact and costs to survivors on a case-by-case basis. Resources are available to survivors with or without them filing a complaint. No upfront costs are charged to any currently enrolled students for University Health Services or the Center for Counseling and Psychological Health, and no fees exist for services in the Dean of Students Office, the Center for Women and Community, Student Legal Services, or by live-in residential staff.

General Education Requirements

Statement

INFO 190S is a 4-credit General Education course that satisfies the R2 (Analytic Reasoning) general education requirements for graduation. The General Education Program at the University of Massachusetts Amherst offers students a unique opportunity to develop critical thinking, communication, and learning skills that will benefit them for a lifetime. For more information about the General Education Program, please visit the GenEd webpage.

General Education Learning Outcomes

The General Education Program has four common objectives that pervade all designations. INFO 190S satisfies the following General Education objectives:

- **Content:** Students will know fundamental questions, ideas, and methods of analysis in computing. In particular, students will learn how to problem solve using a modern programming language, apply programming and problem solving to real world problems.
- **Critical Thinking:** Students will apply and demonstrate creative, analytical, quantitative, & critical thinking through inquiry, problem solving, & synthesis. Students will use critical

thinking skills to solve problems from a computational perspective. As part of the problem-solving process students will use logical reasoning to create algorithms and data structures to develop programs that put their solutions into action. Furthermore, students will investigate aspects of performance and comparisons of equivalent algorithms to draw conclusions on efficiency. Lastly, students will explore and ask questions about real world problems and apply various forms of data analysis to answer these questions.

- **Communication:** Students will develop their writing skills through various assignments that require an articulation of their solution. They will also practice their oral communication by demonstrating their work (i.e., explaining an algorithm or technique) through a recorded video that will be part of assignment submissions.
- **Connections:** Students will connect the material in this course to real world problems such as using programming techniques to predict population size in the future and how that relates to pollution as well as many other issues that exist today. The goal of this course is to intentionally provide a connection between the concepts that are covered and how it impacts the world. This course is not only about learning how to program, it is about how to code to answer questions and to push forward and investigate how to construct solutions to solve problems.

R2 Learning Outcomes

This course will satisfy the R2 learning outcomes. In particular, it will advance student's formal reasoning skills beyond the basic competency level by having them solve programming challenges on a weekly basis using a programming language. This will also increase a student's sophistication as a consumer of numerical information as they must have a fundamental understanding of how a computer represents information (numerical or otherwise) in a discrete environment. Clearly, computer literacy is established in this context and naturally the limits of formal methods and the abuse of numerical arguments will be covered as part of developing programs and solving problems in general.

Schedule

The following is a schedule of topics for this course. This schedule may change as the course progresses based on uncontrolled circumstances, student progress and understanding, or any other reason the instructor determines is a valid reason to make a change.

1. Introduction to Programming
2. Variables and Expressions
3. Types: strings, lists, tuples, and sets
4. Types: dictionaries
5. Functions
6. Functions, Branching, and Control Flow
7. Branching, Detection, and Iteration
8. Iteration and Functional Iteration
9. Strings, Dates, and Lists
10. List Comprehensions and Dictionaries
11. Introduction to Classes

12. Class Interfaces, and Exceptions
13. Files and Plotting
14. Modules and Libraries

Time Management

The university suggests that students spend 3-4 hours of time on a class per credit hour. This is a 4-credit course, so you should plan to spend 12-16 hours a week on this course. In a typical week you will:

- Attend 2 75-minute lectures (unless there is a holiday or class is canceled for any reason)
- Attend 1 50-minute lab (unless there is a holiday or lab is canceled for any reason)
- Complete a lab assignment if one is due that week.
- Complete a homework assignment if one is due that week.
- Complete a programming assignment if one is due that week.
- Complete an exam if one is scheduled for that week.
- Optionally attend office hours.

Code of conduct

- The course staff are committed to providing a friendly, safe and welcoming environment for all, regardless of level of experience, gender identity and expression, sexual orientation, disability, personal appearance, body size, shape, race, ethnicity, age, religion, nationality, or other similar characteristics.
- Please be kind and courteous. There's no need to be mean or rude.
- Respect that people have differences of opinion and that differing approaches to problems in this course may each carry a trade-off and numerous costs. There isn't always a single right answer to complicated questions.
- Please keep unstructured critique to a minimum. Criticism should be constructive.
- We will informally warn you, once, if you insult, demean or harass anyone. That is not welcome behavior. After that we will report your behavior to the Dean of Students office. We interpret the term "harassment" as including the definition in the Citizen Code of Conduct under "Unacceptable Behavior"; if you have any lack of clarity about what might be included in that concept, please read their definition and then ask us for clarification. We don't tolerate behavior that excludes people in socially marginalized groups.
- Private harassment is also unacceptable. No matter who you are, if you feel you have been or are being harassed or made uncomfortable by a member of this class, please contact a member of the course staff immediately (or if you do not feel safe doing so, you should contact the Chair of the Faculty of CICS). Whether you've been at UMass for years or are a newcomer, we care about making this course a safe place for you and we've got your back.
- Likewise, any spamming, trolling, flaming, baiting or other attention-stealing behavior is not welcome.

Academic Honesty

It is very important in all courses that you be honest in all the work that you complete. In this course you must complete all assignments, quizzes, exams, etc. on your own unless otherwise specified. If you do not, you are doing a disservice to yourself, the instructors for the course, the Manning College of Information and Computer Sciences, the University of Massachusetts, and your future. We design our courses to provide you the necessary understanding and skill that will make you an excellent computer scientist. Assignments and exams are designed to test your knowledge and understanding of the material. Plagiarism and academic honesty of any sort may seem like an easy way to solve an immediate problem (which it is not), however, it can have a substantial negative impact on your career as a computer science student. There are many computing jobs out there and many more people working hard to get those positions. If you do not know your stuff you will have a very difficult time finding a job. Please take this seriously.

We will carefully review your submissions automatically and manually to verify that "cheating" has not taken place. If you are suspected of plagiarism we will follow an informal path to determine if academic dishonesty has taken place. If you are found guilty you will receive an F for the course and it will go on your permanent record at UMass. This will disrupt your schedule for completing courses and may lead to you not completing your degree in a timely fashion. You should carefully review the Academic Honesty Policy, Avoiding Plagiarism, and the Academic Honesty Flowchart to understand what academic dishonesty is, how you can avoid it, and the procedure we will follow if you are under suspicion. In general, you should review all documentation described by UMass' Academic Honesty Policy and Procedures.

Specifics for this course:

- Unless otherwise specified, assignments in this course are individual, not group, and direct collaboration is inappropriate. Any group work we will clearly designate as such.
- While we support learning from your peers, the rule of thumb is that any learning will be in your head. Therefore, you should not leave an encounter (in person or electronic) with anything written down (or electronically recorded) that you did not have before. Thus, giving or receiving electronic files is specifically considered cheating.
- Use of materials from previous offerings of this course, no matter the source and even if you are retaking the course, is prohibited.
- We will employ various means, electronic and otherwise, to check for compliance with these course policies. We will pursue sanctions vigorously and the usual sanction we will pursue is an F in the course.

University statement on academic honesty:

Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst. Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed

academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair. Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent

Visit http://www.umass.edu/dean_students/codeofconduct/acadhonesty for more information.