CMSI NNN IDEAS TO CODE: INTRODUCTION TO PROBLEM SOLVING AND PROGRAMMING
FIRST YEAR SEMINAR

3.0 units [4.0 units]
Tuesday, Thursday 9:25 - 10:40 a.m. – Doolan TBD
Dr. Stephanie E. August, Associate Professor

Course Website:	MYLMU Connect
Office Hours:	Wednesday, 1:00-4:00, 5:00-6:00 p.m. *and by appointment.*
Office:	Doolan 201b
Phone:	(310) 338-5973
Internet:	saugust@lmu.edu Put ***IDEAS TO CODE*** in the subject line!!!

Overview

Course Description

This course provides students with the transformative and empowering experience that comes with learning how to program a computer and translate ideas into code. We focus on some of the "Big Ideas" of computing that have impacted the way we present information, solve problems, and connect people. Central questions include: What is an algorithm? How is an algorithm translated into a working software application? Who are the people and what are the technological developments that have made computing ubiquitous? What impact does computing have on society? What constitutes an ethical use of computing technology? Where will computing take us in the future?

Programming projects emphasize animation and artificial intelligence, focus in part on social issues, such as overpopulation or the depletion of natural resources, and can be tailored to the student's major discipline.

Students demonstrate critical thinking and develop oral communication skills through active discussion of program designs and ethical issues related to computing, and through oral presentation of a small research project. They develop writing skills and information literacy through social issues essays, a research paper, and a short reflection paper on computing and the world.

Specific Learning Outcomes:

Upon completion of this course, it is anticipated that the student will be able to:

- Understand the nature and process of software development, including design, implementation, and test
  - Design an algorithm or series of steps that can be used to solve a problem
  - Transform an algorithm into a working program
  - Demonstrate knowledge of basic programming constructs through implementing programs in BYOB (Build Your Own Blocks) and Python Scripting Language
  - Extend an animation built with the 3D animation tool Houdini by building a shelf tool using Python Scripting Language
  - Design, implement, and a test computer program
- Understand the role technology plays in our society
  - Form and support an opinion about various uses of technology

- Engage critically and reflectively in a discussion of
  - Code design, documentation, programming style, and testing
  - Ethical issues related to computing
- Analyze a software program from the perspectives of design and programming style
  - Explain the function of the program at an abstract level
  - Paraphrase the structure of the program at the code level
- Exercise critical thinking in oral discussion and writing.
  - Write a short technical paper
  - Document a software program
- Acquire research skills to
  - Use the on-line library catalog and electronic databases to retrieve books or articles
  - Differentiate between scholarly and popular sources
  - Research and write a short technical paper

Required

Willingness to participate in class discussions and group activities and complete readings prior to lectures. Classroom activities will include both individual and group assignments. No other prerequisites.

Expected Work

Course portfolio containing a representative subset of course assignments.

Six essays on specific social and ethical issues related to course topics, minimum length 250 words. Each essay needs to give a well-formed argument for or against a particular point and be supported by at least one reference beyond the assigned reading. These will be reviewed by the writing instructor, then resubmitted for grading. You will include the best three in your course portfolio due at the end of the term.

Computers and the World Essay: How has your understanding of the world around you has been affected by studying computer science in *Ideas to Code*. Two-pages, 1.5 line-spaced (400-500 word). Draft will be reviewed by the writing instructor, then resubmitted for grading. Both draft with comments from writing instructor and the final essay should be included in the Course Portfolio.

Reflection Research activity completed independently and reported in a two-page paper (excluding images, 500-1000 words in length) and a classroom presentation. References must be cited. Graded deliverables include

Deliverable #1: Topic selection and posting
Deliverable #2: References to be used, a draft of the introductory paragraph, and an outline
Deliverable #3: Draft of activity report with references cited
Deliverable #4: Final report, to be included in the portfolio
Deliverable #5: Oral presentation, to be included in the portfolio and presented in class

Students will submit drafts to the writing instructor for review, then resubmitted for grading. Additional details and interim due dates are found in the Course Portfolio handout. Research activities will be graded on content (50%) and presentation (50%).

Programming:

Seven smaller programming assignments (2 BYOB, 3 Python)

Two larger programming projects (1 BYOB, 1 Python).

Pair programming employed on all programming assignments and projects. Working as a team produces better results and enhances learning. Students are expected to share work

equally and turn in their individual notes and a reflection on the experience along with the work completed by the pair as a team.

Each team will make an oral presentation (with visuals) to the class on their project; reviews will include design review, code review, and demo.

Software is a programmer's version of an essay. Working code is not enough. Programs and tests must be clearly documented, observe established programming standards, exhibit good design and degrade gracefully.

Completion of all book problems related to the assigned readings is recommended.

Exams

Quiz, midterm.

Text and Required Materials

BYOB Tutorials. Available online: byob.berkeley.edu

Blown to Bits: Your Life, Liberty, and Digital Happiness After the Digital Explosion. Hal Abelson, Ken Ledeen, and Harry Lewis. Addison-Wesley. 2008. Available online: http://www.bitsbook.com/. Referred to as "B2B". This is our source for social issues.

Scratch: programming for all. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. *Commun. ACM* 52, 11 (November 2009), 60-67. DOI=10.1145/1592761.1592779 http://doi.acm.org/10.1145/1592761.1592779

How to How to Think Like a Computer Scientist: Learning with Python. Allen B. Downey. O'Reilly Media. 2012. Available online: http://www.greenteapress.com/thinkpython/ Referred to as "LP"

Supplementary materials as posted on Blackboard or handed out in class.

Required technology:

- Access to a computer running BYOB, Python, and Houdini Apprentice (students will learn how to download and install this software during the first week of class)
- Access to course information on *MyLMU Connect*. Students are expected to inspect the course Blackboard frequently for announcements, updates, assignments, and documents, and to use Blackboard to communicate among class members.
- *LionShare* or similar file sharing software to make student files available to the instructor and other students.

Useful technology:

- Laptop for in-class activities. Not required; these group activities simply require someone in each group to bring a laptop to class.

Additional References

Algorithmics: The Spirit of Computing. David Harel with Yishai Feldman. 3rd edition. Addison-Wesley, 2004.

Computer Science: An Overview. J. Glenn Brookshear. 9th, 10th, or 11th ed. Addison-Wesley, 2010-2012.

Houdini Apprentice Edition.
http://www.sidefx.com/index.php?option=com_content&task=view&id=589&Itemid=221

Houdini Tutorials.
http://www.sidefx.com/index.php?Itemid=132&id=14&option=com_content&task=blogsection

Scratch: programming for all. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. *Commun. ACM* 52, 11 (November 2009), 60-67. DOI=10.1145/1592761.1592779 http://doi.acm.org/10.1145/1592761.1592779

Other vetted sources of information are available, both in the library and on the web. Students are encouraged to take advantage of these to gain a deeper understanding of the topics covered in class and in the text.

Grading

Your final grade will be weighted as follows:

Programming Assignments	20%
Course Portfolio	35%
Social Issues Essays	8%
Research Activity and Paper	5%
BYOB Project	10%
Python Project	10%
Computers and the World Reflection	2%
Oral Presentations and Participation	15%
Oral Presentation on the Research Activity in the Course Portfolio	
Software Design Reviews	
Software Code Reviews	
Project Demos	
Classroom Individual and Group Activities	
Information literacy tutorial and assessment	10%
Quiz	5%
Midterm	15%

Programming assignments, course portfolio materials, and oral presentations will be graded on content (50%) and presentation (50%).

Students will participate in several group activities in the classroom over the course of the semester. Students will be graded both on the group product and on their individual contribution to the group effort. Students who miss these activities will receive no credit for the group product. Make sure your name is on each group report.

Toward the end of the semester each student will submit a portfolio of a representative sample of their work, a final copy of their programming assignments and related documentation, and certain other written assignments. Students can revise and improve assignments submitted during the semester and include the improved versions of their work in the portfolio. These will be graded more closely, as described in the Course Portfolio handout. Note that this portfolio includes a research activity and a brief in-class presentation on a topic that the student selects from a list of suggested activities. Related homework assignments help students stay on track for completing the activity.

An incomplete will be granted only when the student requesting the incomplete has completed 80% of the coursework, and has at least a B average in the coursework completed.

Assignments are due at the beginning of class on the due date and should be uploaded to MyLMU|Connect before class starts. Papers not submitted by 10 minutes after class has started are considered late. Late assignments are not accepted unless otherwise noted.

Refer to the ***Teaching Philosophy and Course Policies*** handout for additional information.

Ideas to Code Schedule: Topics, Readings, Assignments

week	topic	reading	assignment due
1	Computing in our world BYOB: Problem Solving I	B2B: ch. 1	
2	Computers and Privacy BYOB: Building Tool Blocks Writing Tutor: Structure of Social Issues Essays, Citing References	B2B: ch 2 BYOB Building Tool Blocks Tutorial	Social Issues Essay #1
3	BYOB: Lists and Higher Order Functions Librarian: Online Catalog and Electronic Resources, Differentiating Between Scholarly and Popular Resources	BYOB Lists and Higher Order Functions Tutorial	Programming Assignment #1
4	Amination I, Design Tools BYOB: Recursion Quiz	BYOB Recursion Tutorial	Social Issues Essay #2 Research Activity Deliverable #1 (post)
5	Programming Standards and Documentation Secrets of Electronic Documents Writing Tutor: Correctly Citing Sources, Plagiarism	B2B: ch.3	Programming Assignment #2
6	Software Engineering and Testing Web Search	B2B: ch. 4	Social Issues Essay #3 Research Activity Deliverable #2: refs, intro, outline
7	Python: Problem Solving II, Variables Expressions Statements Cryptography	LP: ch. 1, 2 B2B ch. 5	BYOB Project due
8	Python:Functions I, Conditionals	LP: ch. 3, 4	Social Issues Essay #4
9	Animation II Python: Functions II, Iteration Bit Ownership: Copyright Issues	LP: ch. 5, 6 B2B: ch. 6	Programming Assignment #3 Research Activity Deliverable #3 (draft) Research Activity oral presentations, weeks 9 - 14
10	Python: Strings, Video Game	LP: ch. 7, 8	Social Issues Essay #5
11	Artificial Intelligence I Python: Lists Frontiers of Digital Expression	LP: ch. 9 B2B: ch. 7	Programming Assignment #4
12	Python: Modules and Files Midterm	LP: ch. 10	Social Issues Essay #6
13	Artificial Intelligence II Python: Recursion and Exceptions	LP: ch. 11	Programming Assignment #5
14	Python: Dictionaries	LP: ch. 12	
15	Digital Censorship The Future	B2B: ch. 8	Python Project due Portfolios due

ding assignments will be refined as the term progresses.

Teaching Philosophy and Course Policies

Dr. Stephanie E. August

Philosophy

The instructor guides, the student explores. My job is to show the students what to learn and how to master it. The student's job is to explore the paths laid out before them. I assume that students are taking the class because the subject interests them and expect the students to take the initiative in learning the material. Everything we learn will be useful later, so it is important to be exposed to many different ideas.

I expect all students to contribute to the learning experience and not simply wait to be taught. In a graduate course, students come to class prepared to discuss the topic at hand, relate it to their previous experience, and project how it will be used in the future. Although undergraduates might think they have little direct experience with the topic, they are encouraged to find evidence of it in their life experiences and current news articles. All students are advised to transfer knowledge from other disciplines wherever possible.

Students are always welcome to discuss course material with the instructor, but they are also expected take responsibility for mastering the course material and to seek out additional references for amplification and clarification of course concepts. The course description provides several additional references to use as a starting point.

Expected Work

Course-specific expectations are noted in the Course Description. In addition, I have the following general expectations.

Students are expected to take responsibility for mastering course material, rather than expecting to be provided with all the answers. If an assignment is unclear, the student should discuss the assignment with the instructor, but is also expected to seek out published resources related to the assignment.

Students are responsible for all the material in the assigned readings, whether or not it is covered in class, and for all material presented in class, whether or not it is in the assigned readings. Students are expected to complete the assigned reading prior to lecture and to participate in class discussions.

Students are expected to take responsibility for keeping track of deliverables and due dates throughout the semester. Students are expected to turn in materials according to the schedule distributed by the instructor at the beginning of the term, unless the instructor explicitly issues an updated course schedule; students should not expect the instructor to remind them of due dates. Late assignments are not accepted, except as noted in the Course Description or announced in class.

Course information will be published on MyLMU Connect, formerly known as Blackboard (BB). Students are expected to:

- inspect the course MyLMU Connect site frequently for announcements, updates, assignments, documents
- use the course MyLMU Connect site to communicate among class members
- make certain they receive email from the email address listed for them on MyLMU Connect

Email Communication

At times I will communicate with the entire class using campus email systems, or with individual students using MyLMU Connect or the student's Lion mail account on PROWL. **It is essential** that you regularly check your lion.lmu.edu email address or forward your lion account email to your preferred email address.

Participation

Students are expected to be active participants throughout the entire class and to contribute to the quality of the discussion. Please note that the frequency with which a student speaks in class is not a key criterion for effective class participation. The classroom should be considered a laboratory in which students can test their ability to convince their peers that they have approached complex problems correctly and that their approach will achieve the desired results.

Criteria that we use to measure effective class participation include:

1. Is the comment clear and relevant to the current discussion?
2. Does the student support the comment well using case facts and tools developed in the class?
3. Does the student explore all the implications and importance of the comment?
4. Is the comment insightful? Does it broaden the discussion and clarify the issues?
5. Are comments complete and concise (does the comment cover the point as well as possible in a few words as possible)?

An average comment satisfies 1 and some of 2. A good comment satisfies 1-3. An excellent comment satisfies 1-5. Class participation represents a major component of your grade in this course (20-30%).

Before asking "Will this be on the test?" refer to the section on *Expected Work* above and contemplate whether the question meets these criteria for effective class participation.

Students are not allowed to say 'I don't know' in this class when asked a question. A student is not required to know, but is expected to think. So if I ask you a question and you don't know the answer, you are responsible to think of an answer, to guess, to speculate, to wonder aloud.

Questions referring to material that was covered when the student asking the question was absent will be answered (or not) in class at the instructor's discretion.

While participation is should be the norm, students are cautioned against over participating. If you have made several contributions already, let someone else have a turn, and avoid calling out answers to questions directed toward other students.

Grading

Your final grade will be weighted as noted in the Course Description handout.

Graduate-level final grades are assigned as follows:

A-/A	Superior	Student has mastered the material and presents it in a professional manner.
B-/B/B+	Satisfactory	Student has a good grasp of the material and presents it clearly.
C+/C/C-	Poor	Student did some work, but does not have a strong grasp of fundamental concepts.
F	Fail	Student failed to learn fundamental concepts.

Graduate students must earn at least a B in each 500-level course for it to count toward their degree. 500-level courses in which the student receives a B- or lower will need to be repeated for credit.

Undergraduate-level final grades are assigned as follows:

90- 100%	A-/A	Student has mastered the material
80- 89%	B-/B/B+	Student has a good grasp of the material.
70- 79%	C-/C/C+	Student has a basic understanding of the material.
50- 69%	D	Student did some work, but failed to learn fundamental concepts.
0- 49%	F	Student did an insufficient amount of work to satisfy course requirements.

Unless otherwise stated for a specific assignment or deliverable, half of the grade will reflect *content*, and half of the grade will reflect *presentation*.

An incomplete will be granted only when the student requesting the incomplete has completed 80% of the coursework, and has at least a B average in the coursework completed.

Assignments

Submitted assignments reflect your attitude toward learning. Written assignments may be handwritten or typed. All are expected to be neat and legible. *Homework and papers with multiple spelling and/or grammatical errors or not representative of college-level work will be returned ungraded.* Find yourself a good spelling and grammar checker and/or a trusted human editor, if you have difficulty with English spelling and/or grammar. The Academic Resources Center in Daum Hall offers such a service, both on an appointment and a drop-in basis.

Student's name and the assignment number and due date must appear in the upper right corner of the first page of the assignment. Answer questions with complete sentences; include the question in the answer and show your reasoning.

Staple pages of assignments submitted on paper. Points will be deducted for loose pages.

Exams

Once a quiz or exam has begun, it will be assumed that anyone leaving the room is finished with the exam. Have adequate supplies (paper, pens, pencils, tissues) on hand, and take care of personal needs before coming to class. No makeup quizzes or exams will be given, except as noted below.

If your job, team sports, or other commitments require you to travel during the term, special arrangements should be made prior to a missed class for submitting assignments, receiving assignments and handouts, or rescheduling quizzes or exams. If you miss a quiz or exam without making prior arrangements, I will enter a grade of 0 for the quiz or exam, and no makeups will be allowed.

Electronic Devices

Turn off and put out of sight all electronic devices other than laptops used for taking notes in class. Reading emails, completing homework assignments, or searching the internet for anything that will not augment the classroom experience for the entire class (students and instructor) should be avoided altogether. The distractions they cause disrupt class and usurp precious class time. A repeat offender may lose credit for the day's work.

Academic Honesty and Integrity

Students find it helpful to discuss approaches to assignments and projects with their classmates. However, unless an assignment has explicitly been declared a team assignment, each student is expected to complete and write up the assignment or project component on his or her own. When an approach has been discussed in a group, each student should clearly note on the copy of the work that he or she submits what kind of collaboration occurred, and the name of each collaborator. Researching a problem on the internet is considered to be collaboration and should be noted on the assignment. Relevant URLs should be noted on the assignment. Cheating on assignments by failing to note collaboration or not writing up the assignment on an individual basis, cheating on examinations, plagiarism, falsification of data, and related violations of LMU standards of honesty and integrity are not tolerated. Students who commit such offenses will receive a failing grade for the assignment or exam and/or a failing grade for the course, as well further disciplinary action.

Students are expected to understand what plagiarism is and avoid all forms of it. The website http://libguides.lmu.edu/plagiarism explains the plagiarism and discusses techniques for avoiding it. The page includes links to several informative tutorials.

Academic dishonesty will be treated as an extremely serious matter, with serious consequences that can range from receiving no credit for assignments/tests to expulsion. It is never permissible to turn in any work that has been copied from another student or copied from a source without properly acknowledging the source. It is your responsibility to make sure that your work meets the standard of academic honesty set forth in the "LMU Honor Code and Process" in the 2011-2012 Bulletin pages 49.

Americans with Disabilities Act

Students with special needs as addressed by the Americans with Disabilities Act who need reasonable modifications, special assistance, or accommodations in this course should promptly direct their request to the Disability Support Services Office. Any student who currently has a documented disability (physical, learning, or psychological) needing academic accommodations should contact the Disability Services Office (Daum Hall Room 224, 310-338-4535) as early in the semester as possible. All discussions will remain confidential. Please visit http://www.lmu.edu/dss for additional information.

Survival

Eat before you are hungry, drink before you are thirsty, and sleep before you are tired, and you will have energy left to celebrate the completion of the course and enjoy the semester break. Eating, drinking, and sleeping are to be done outside the labs and classrooms. Repeat offenders will find it difficult to complete the course, since they will be asked to leave the classroom and will lose their lab privileges.

Portfolio

I. Overview

A course portfolio, worth 50% of the course grade, contains a representative subset of course assignments. The objective of creating a course portfolio is to provide the student an opportunity to demonstrate mastery of course concepts without the time constraints or pressure of an exam and with the benefit of additional time to work with the material and receive feedback on assignments. Essays on social issues, a short research paper, two programs, and a short reflection paper on computers and the world comprise the portfolio. The majority of these will be extensions of class assignments and activities.

The completed portfolio is due at the last week of class. Portfolios submitted after that date will be penalized for each day they are late.

Section II of this document discusses the content of the portfolio. Section III reviews the rubric used to assess your work, and section IV discusses the disposition of the portfolio once it is graded. Section V lists relevant due dates.

Students often require extra assistance with the programming tasks. Consider working on these early and often to allow adequate time to resolve any challenges that arise.

II. Content

All material should be secured in a report folder that clearly shows on the cover the name of the student, the title of the course, and the date the portfolio is submitted. Items should be included in the order listed here. Tabs or dividers should identify and separate the sections.

Section 1: Social Issues Essays

During the semester, students compose essays on a variety of social issues related to computing sciences as part of graded homework assignments. The objective of these essays is to reflect on the association between computing sciences and daily life. This section of the portfolio consists of ***three*** of these essays. These can either be graded and revised essays or new essays. *For full credit,* each essay must

- address a specific social issue question related to a course topic
- identify the question addressed
- relate the response to information covered by the course
- be a minimum of 250 words in length
- cite at least one reference beyond the text to support the points made in the essay
- include the full citation of the article(s) at the end of the essay.

Section 2: Research Activity

The instructor will suggest research activities related to course topics, or the student can propose a topic related to their interests and the course material. Select and complete one activity and write a two-page paper as described in the activity. If the activity involves programming, the paper should include a description of the activity and the design of the

code. The code itself and an example of the code executing or graphics drawn by the code should be added as an appendix. Cite the references used and include them in a brief References section at the end of the paper. Interim due dates for each of the following tasks are found in *Section V. Due Dates*.

- Indicate the research activity selected on the *Research Activity* wiki of the course MyLMU|Connect (see Tools/Wiki Tool/Site Navigation/Research Activities).
- Begin a discussion thread for your topic under the *Research Activity Presentations* forum on MyLMU|Connect. Include a sentence or two about why you selected the topic.
- In the interest of diversity, each student is encouraged to select a unique activity. No more than two students can sign up for any specific activity.
    - A student must obtain prior written approval of the instructor to select a topic other than those included in the posted Research Activity list, or one previously selected by two other students.
- Submit a *draft* of the paper to the writing instructor for review. Include properly cited references.
- Make a 5-minute presentation to the class on your research. Each student will be assigned a presentation date between the **9th and 14th week of class**.
- Post the slides for your presentation to your research activity thread on MyLMU|Connect within 24 hours of making your presentation
- Submit the final version of the paper and the draft version with the writing instructor's feedback with the portfolio.
- Include a 4-up landscape hard copy of your slides in the portfolio; double-sided is fine.

The selection of activity, paper draft, and presentation will each count as one homework assignment.

Section 3: BYOB Project

Design, document, and test a short program in BYOB that either calculates a value or draws something based upon user input. As a minimum use the following constructs or parts of BYOB:

- Variables (2+)
- Control components (3+) , e.g.,
    - Loop (repeat or forever)
    - Conditional (if, if...then…, if..then...else…)
    - Event control block (with curved top)
- Operators (3+) , e.g.,
    - +, -, >, pick random

The source code can be represented with a screen shot of the program taken in the BYOB development environment.

Ideas to Code Portfolio

Include in the portfolio:

- A README file containing brief narrative explanation of what the program does, which files are needed to run the program (the *inventory*), how it does it, how to test the program, and how to run the code .
- A diagram that explains graphically how the program works. A flow chart or UML sequence diagram with an associated use case diagram would be most appropriate.
- A list of primitives needed to solve the problem.
- The source code for the program.
- Screen shots of your program executing.
- A list of tests a user can run to verify that the program runs properly, expressed either as input/output pairs or user action/observation pairs.

These items should be compiled and presented in the appropriate section of the course portfolio in such a way that they tell the story of the project in a coherent and cohesive manner.

Section 4: Python Project

Design, document, and test a short program in Python that either calculates a value or draws something based upon user input. Your program should include as a minimum use:

- Variables (2+)
- Control components (3+) , e.g.,
  - Loop (repeat or forever)
  - Conditional (if, if...then…, if..then...else…)
  - Event control block (with curved top)
- Operators (3+) , e.g.,
  - +, -, >, pick random
- Two user-defined function definitions

The source code must be included as a standalone document can be represented with a screen shot of the program taken in the BYOB development environment.

Include in the portfolio:

- A README file containing brief narrative explanation of what the program does, which files are needed to run the program (the *inventory*), how it does it, how to test the program, and how to run the code .
- A diagram that explains graphically how the program works. A flow chart or UML sequence diagram with an associated use case diagram would be most appropriate.
- A list of primitives needed to solve the problem.
- The source code for the program. (You will need to save your code to a separate file and print it out.)
- Screen shots of your program executing.
- A list of tests a user can run to verify that the program runs properly, expressed either as input/output pairs or user action/observation pairs.

Ideas to Code Portfolio

These items should be compiled and presented in the appropriate section of the course portfolio in such a way that they tell the story of the project in a coherent and cohesive manner.

Section 5: Computers and the World

Write a two-page 1.5 line-spaced (400-500 word) essay on how your understanding of the world around you has been affected by studying computer science in *Ideas to Code*. The idea is to examine the importance computer science plays in our everyday lives by focusing on a single *event* in which a computer or computing plays a significant role. The event can be any of the following:

- A movie or PBS special (e.g., The Net, A.I., I.Robot, Terminator, Jurassic Park).
- A current event or news article.
- An aspect of your daily life.

The content of your essay you will be the same in each case. In your essay,

- Briefly summarize the event you have selected and identify the role computers or computing plays in the event.
- Compare what your understanding of the event would have been before taking CMSI 182 to what your current understanding is. The idea is to examine how your view of the world has changed by being more aware of computer science.
- Identify the single most important idea you have taken from the class
- Identify the course topic that you think will be the most useful to you in the next five years as you continue in school or venture out into the world to work.

You will submit a draft to the writing instructor for review by the writing instructor, then include the final essay for grading. Both draft with comments from writing instructor and the final essay are to be included in the Course Portfolio.

III. Rubric

The objective of the portfolio assignment is for you to submit a professionally presented set of work that represents your mastery of course material. The portfolio work must meet a higher standard than your other assignments because you will have had time to review the feedback received on earlier assignments and have learned more about the subject matter by the end of the term. Your portfolio grade will be determined by both the content of the work and your presentation of the work. A well-presented set of incomplete or incorrect assignments will not receive a high grade, but neither will a set of stunningly correct and thorough assignments that are illegible or poorly presented.

Your written portfolio work will be graded using the following criteria:

Content	50%	Includes the requested information; substantive, reflecting depth of thought
Presentation	50%	Ideas are stated clearly and flow smoothly, grammar and spelling are correct. Errors are neatly corrected.

Ideas to Code Portfolio

For full credit on all written work such as social issues responses, research activities, computers and the world essay, cite highly regarded references that provide examples of the topic being discussed or support your view and include a full citation for each reference at the end of each piece. The full citation must include author, title, publication, publication, volume, issue, and date. If the reference was found on the web, provide the full citation plus the URL and the date the article was accessed. The idea is to provide sufficient information for someone to locate and read the article for more information, even if the web link becomes inactive.

Technical work will be graded along these criteria:

Design	30%	Clear, flexible, and easily maintained; elegant and innovative; applies proper separation of concerns; satisfies the "one change, one place" property
Functionality	30%	Works as requested; meets all requirements; produces correct answers/results; performs in a reasonable amount of time; includes tests that demonstrate correct behavior
Naming	20%	Clear and consist; names correspond to roles, types, or actions
Documentation	20%	README or overview material provided; comments abundant in code; information is genuinely useful

The completed portfolio is due the last week of class.

IV. Portfolio Disposition

Portfolios will be available in my office after grades are turned in (usually the Wednesday following finals) until the third week of the following fall semester, after which the contents will be discarded and the folders recycled, unless you have made other arrangements to pick up your portfolio. If you know that you will not be picking up your graded portfolio, indicate that on the title page of your portfolio.

V. Due Dates

Portfolio Component	***Due date***
Social Issues Essay #1	week 2
Social Issues Essay #2	week 4
Indicate research activity selected on the *Research Activity* wiki	week 4
Begin a discussion thread for your research activity	week 4
Social Issues Essay #3	week 6
Turn in research activity draft paper by start of class	week 6
BYOB Project	week 7
Social Issues Essay #4	week 8
Present research activity (5 minute presentation)	week 9-12
Social Issues Essay #5	week 10
Social Issues Essay #6	week 12
Python Program	week 15
Submit completed portfolio	week 15