CS 100: Principles of Computing (3 credits)

Kevin Molloy

Spring 2018

1 Basic Course Information

1.1 Prerequisites: None

1.2 General Education

Fulfills Mason Core requirement in Information Technology (ALL).

1.3 Meeting Times and Locations

Section	Days	Time	Professor	Location
001	T/R	12:00-1:15pm	Molloy	Arts and Design 2026
002	T/R	9:00-10:15am	Molloy	Arts and Design L008

1.4 Instructors

Name	Kevin Molloy
Office	Engineering 4101
Email	kmolloy1@gmu.edu

1.5 Teaching Assistants

Call Me	Name	Email	
Ramandeep	Ramandeep Kaur	rkaur22@gmu.edu	GTA
Nnamdi	Nnamdi Ojibe	nojibe@masonlive.gmu.edu	UTA
Maryam	Maryam Alarcon	malarcon@masonlive.gmu.edu	UTA

1.6 Required Texts

- *The Pattern On The Stone: The Simple Ideas That Make Computers Work* by W. Daniel Hillis. Basic Books (1999). Available at the GMU bookstore and at Amazon.

- *How to Think Like a Computer Scientist: Learning with Python 3* by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers; October 2012 Edition. This free online textbook (available by clicking on the title or clicking here) will cover the basic Python concepts we will discuss in class.

1.7 Computing Resources

It is required you will have access to a computer with the ability to browse the internet and install programs like Python. Some university labs provide computers for this purpose but most students find it more convenient to have a personal computer. Lecture time will be devoted to installing necessary programs when the need arises. If that it will be difficult for you to gain access to computing resources, contact the course staff.

1.8 Communication

- **Blackboard** is our central course web site. The announcements, discussion board, videos, and documents posted there are part of the required reading for the course.
  - All instructors and TAs can view all material on Blackboard
  - Use public posts on Blackboard to discuss HW, and other material related to the course.
  - Use private posts on Blackboard for questions involving your own work or code.
  - Blackboard is used to submit assignments and disseminate grades
  - **Mail the professor or TA directly** if you have logistic or personal issues to discuss such as setting up an appointment outside of office hours, if a health problem arises, or if you have a personal emergency.
- **Office Hours** are listed for all staff on Blackboard under course (Resources/Staff). No appointments are required to attend office hours: staff will be at the listed location at the given times with doors open ready to help. In addition to attending the regular meeting times, you are strongly encouraged to visit course staff during office hours to further your understanding of the material: we are here to help you learn.

2 Course Description

2.1 Goals

This course is intended to help students learn to think in the manner necessary to fully grasp the nature and power of the digital world around us. The early era of the Internet and the personal computer led to the need for "computer literacy." Now, the changing nature of our global society requires that students learn new ways to think about problems and how to solve them, regardless of students' specific fields of endeavor. Through this course, students will explore major issues related to the "big ideas" of computational thinking (namely, (i) Creativity, (ii) Abstraction, (iii) Data, (iv) Algorithms, (v) Programming, (vi) Internet, and (vii) Societal Impact), as well as how these issues will impact their future lives.

Topics to be covered include.

- Representation of ideas with bits
- Basic boolean logic
- Devices to implement logic functions
- Programming Languages for data and action
- Primitive Operations and Abstraction
- Algorithms for work and play
- Communication between machines
- Computing Security Concepts
- Basic data analysis
- Impact of automation and communication on human societies
- Human and Machine Intelligence

2.2 Learning Outcomes

By the end of the semester, a passing student will be able to carry out the following types of activities:

1. Students will be able to use technology to locate, access, evaluate, and use information, and appropriately cite resources from digital/electronic media.

2. Students will understand the core IT concepts in a range of current and emerging technologies and learn to apply appropriate technologies to a range of tasks.
3. Students will understand many of the key ethical, legal and social issues related to information technology and how to interpret and comply with ethical principles, laws, regulations, and institutional policies.
4. Students will understand the essential issues related to information security, how to take precautions and use techniques and tools to defend against computer crimes.
5. Students will recognize the significance of the "big ideas" of computational thinking.

3 Coursework

3.1 Lectures

During lectures we will discuss computing concepts and instructors will provide demos of programming relevant to assignments. You are expected to complete the reading assignments before class, so that lecture time can be spend increasing your understanding. Occasionally some work will be done in class which students submit and will contribute to the student's overall grade in the class. The days in which in-class work will be done **will not be announced in advance**.

3.2 Reading

Readings from the textbook(s) relevant to each lecture are listed in the schedule, and you are expected to read this material before the associated lecture. Additional reading material and videos will be provided to supplement the textbook which will be posted on Blackboard.

3.3 Homework Assignments

Homework will be a mixture of problem solving, writing English responses to questions, and writing code. Some HW will be individual effort and other HW may be done in groups of a limited size. All HW must be submitted in electronic form to Blackboard by posted deadlines. The exact nature of what should be submitted will be described in each HW. Some HW assignments will be relatively short and have a short turn-around time for submission (1 week) while other assignments will be longer with a turn-around reflecting the effort (2-3 weeks).

Homework will be submitted electronically on Blackboard. Students may submit paper copies to get additional feedback from the grader.

3.4 Exams

There will be a series of mini-exams which last 30 minutes during lecture time. There will also be a comprehensive final exam at the end semester. Refer to the schedule for dates of the exams.

3.5 Bonus Credit

Bonus credit will be awarded based on participation in class discussions. Students may elect to sit in the first few rows of the room ("hot seats") and answer questions. Reasonable effort on answering questions in class will garner class participation credit. Participation points may also earned for involvement in the class discussion board such as giving suggestions to students with questions (but not revealing answers wholesale). The highest point winner at the end of the semester will receive a 3% bonus to their overall score in the course. All other students will receive a bonus proportional to the highest point winner. For example, someone tied with the highest point scorer will also receive a 3% bonus while someone with half the participation points will receive a 1.5% bonus.

3.6 Class Schedule

Week	Date	Topic	Reading Material	Due
1	Jan 23	Overview	Pattern Ch 0	
	Jan 25	Primitive Operations; Algorithms	Pattern Ch 1	
2	Jan 30	Bits & Bytes, Boolean Logic	Pattern Ch 2	
	Feb 1	Bit Interpretation		HW 1
3	Feb 6	Python, Turtle Graphics	Pattern Ch 3 ;Think Ch 1,2 & 3	
	Feb 8	Review, **Mini-exam 1**		HW 2
4	Feb 13	Abstraction using functions	Think Ch 4 & 5	
	Feb 15	Programming loops & conditional	Think Ch 6 & 7	
5	Feb 20	Computability	Pattern Ch 4	
	Feb 22	Python Lists	Think Ch 11	HW 3
6	Feb 27	Python Lists		
	Mar 1	Review **Mini-exam 2**		
7	Mar 6	Algorithmic Complexity	Pattern Ch 5	
	Mar 8	Search Algorithms		HW 4
8	Mar 13	**No Class - Spring Break**		
	Mar 15	**No Class - Spring Break**		
9	Mar 20	Parallel Computing	Pattern Ch 6	
	Mar 22	Encryption	Pattern Ch 7	
10	Mar 27	Operating Systems		HW 5
	Mar 29	The Internet & HTML		
11	Apr 3	Review **Mini-exam 3**		
	Apr 5	Putting Stuff on the Web		
12	Apr 10	Simulation and Randomness	Pattern Ch 8, ML Tutorial	HW 6
	Apr 12	AI and Machine Learning	Plan for SPAM, Frontline videos	
13	Apr 17	Privacy w/Machine Learning		
	Apr 19	Automation	Pattern Ch 9	HW 7
14	Apr 24	Intellectual Property		
	Apr 26	Review **Mini-exam 4**		
15	May 1	Putting it all Together		
	May 3	Review		
–	May 10	Final Exam (sect 001)	10:30am -1:15 pm	
	May 10	Final Exam (sect 002)	7:30 am to 10:15 am	

4 Grading Policies

4.1 Components

Final grades will be determined by scores obtained on homework assignments, in class work, mini-exams, final exam, and term paper. If circumstances require it, the grading scale may be adjusted, generally in the students' favor. The approximate weighting of each component is given below.

Component	Count	Weight
Homework Assignments	5-7	40%
In-lecture Work	5-7	10%
Mini-Exams	3-4	30%
Final Exam	1	20%

4.2 Final Grade Determination

Final grades will be assigned **without rounding** according to the following criteria. It is a 10-point scale per letter grade, with the upper and lower 2% of each 10% earning a + or -.

Percent	Grade	Percent	Grade	Percent	Grade	Percent	Grade
>= 98	A+	89-88	B+	79-78	C+	69-60	D
97-92	A	87-82	B	77-72	C	<60	F
91-90	A-	81-80	B-	71-70	C-		

4.3 Grading Disputes

Address grading issues with the grader first which is either the UTA or GTA and will be included as part of grading feedback. Disputes should be made respectfully either in person or via e-mail. If it is not possible to reach a resolution, the professor may be contacted by the grader or student to resolve the dispute.

If you have not initiated contact within **1 week** after receiving a grade, the chance to contest the grade has closed.

4.4 Homework Deadlines and Late Submission

- Homework is due via electronic submission by the beginning of class on the specified due dates (usually Thursday class). You can submit work to BlackBoard as many times as desired. Only the last submission will be graded.

- Work will **not be accepted** more than 4 days after the assigned deadline.

- Late work will be accepted with a -10% penalty per day late.

  Example: HW is due on 10/10 and Jafar hands in his assignment on 10/12. Jafar would have scored 83 but instead gets `83 - 10*2 = 63`.

Iago wants to hand in his assigment on 10/15 but it will not be accepted as it is too late (> 4 days late).

- On-time submissions will generally be graded and available a week from submission, though exceptional circumstances may cause delays. Late submissions will be graded in as timely a fashion as schedules allow.
- Contact the professor as soon as possible if health or family emergencies occur that interfere with completing course work.

4.5 Exam Policies

- **Your Mason ID is required** for all exams. You may be asked to show the ID to verify your identity on handing in your exam.
- Missing an exam results in a zero score and make-up exams will be considered only in situations involving death and near death. Proof of such circumstances will be required for a make-up to be considered.

5 Academic Honesty and Collaboration

5.1 PRIME DIRECTIVE

PRIME DIRECTIVE: Be able to explain your own work including homework code and exam solutions.

Nearly all cheating in programming can be averted by adhering to the PRIME-DIRECTIVE. Students may be asked at any time to explain code or exam solutions they submit. Inability to do so will be construed as evidence of misconduct. More specific guidelines are given below.

5.2 Thou Shalt Not

For the purposes of this course, the following actions constitute scholastic misconduct (cheating):

- Directly copying someone else's solution to a homework problem, including student solutions from a previous semester
- Directly copying an answer from some outside source such as the Internet or friend for a homework problem
- Making use of an Instructor Solution manual to complete homework problems

- Paying someone for a homework solution or submitting someone else's work as your own
- Posting solutions to any web site including our course web site
- Collaborating or copying someone else's answer during an exam
- Aiding or abetting any of the above
- Witnessing any of the above and failing to report it the instructor immediately

Refer to the following links for additional information.

- Computer science specific policies on scholastic dishonesty:

  `http://cs.gmu.edu/wiki/pmwiki.php/HonorCode/CSHonorCodePolicies`
- University Honor Code: `http://academicintegrity.gmu.edu/honorcode/`

5.3 Penalties

Any instance of misconduct that is detected will be referred to the honor board and will likely result in failing the course. Be advised that the teaching team will be employing **electronic means to detect plagiarism**. This is extremely easy with computer code so keep your nose clean.

5.4 Fair Collaboration

The purpose of this course is to learn about programming and learning from one another is a great help. To that end, the following actions **will NOT be considered cheating in this course.**

- Talking to other students in the course about HW problems and informally describing how a problem may be solved.
  - Be **very careful** as you do this that you do not share any sort of code as this will be detected.
- Getting or giving help fixing a small bug or two: a second set of eyes is a great boon to finding that misplaced semicolon that is preventing your code from compiling.
- Searching the Internet for alternative presentations of a programming concept.

- When unsure whether collaboration is fair or not, stop the activity until it can be cleared with instructor.

At all times keep the PRIME-DIRECTIVE in mind when studying with another student. The above collaborations should be limited to getting someone over a hurdle, not carrying them across the finish line.

About half your grade will depend on programming projects. Doing them individually prepares you for the exams in which no collaboration of any kind is allowed.

6 Additional Policies

6.1 Programming Assignments

Since this is a programming course, some special policies will be in effect.

- Programming assignments will be submitted to Blackboard at the specified dates. You may submit assignments to Blackboard as many times as you like: only the most recent submission will be graded (up to the deadline).
- Back up your program regularly. Keeping backup copies of your work is your responsibility. Losing work due to a disk failure, accidental deletion, etc., is not a justification for late submission. Technologies like Dropbox automate the task of backing up which can make life much easier and safer. Submitting occasional early versions to Blackboard is also wise. Should last-minute problems happen such as accidental deletion, you will at least have some of your work to show. No code to submit means no credit.
- Submitted code that does not compile or run error-free will receive little to no credit.
- When test cases are provided by the instructor use them and make sure your code passes all tests.
- Familiarize yourself with how plagiarism works with programming as described in other parts of the syllabus.
- Keep an untouched copy of your final code submission. It is important that you not touch your programs once you have made your final submission. If there are any submission problems, consideration for credit will only be given if it can be verified that the programs were not changed after being submitted.

6.2 Behavior and Accommodations

Students are expected to maintain a high level of civility for all participants in and out of class meetings. This includes respecting the beliefs of participants of all genders, ethnicities, and social backgrounds. Harassment of any type will not be tolerated and failure to behave in a respectful manner will result in referrals to University Counseling or the Office of Student Judicial Affairs. Any instances of sexual harassment will be reported to the Office of Equal Opportunity according the following policy:

`http://universitypolicy.gmu.edu/1202gen.html`

Observance of religious events will be accommodated for students of any faith.

All possible accommodations will be made for students with disabilities. Please contact Disability Services (`http://ods.gmu.edu/`) and the instructor for further information.

7 Guidelines for Success

- Most homework problems have analogs that are (1) discussed in lecture or (2) described in an example problem in the textbook. This should encourage you to

  - Attend lecture and take notes
  - Read the content of each section of the textbook carefully
  - Analyze the example code distributed by the instructor

  Most successful students do all of these.

- Seek help from the instructor or teaching assistant when a concept is unclear. If you blow a problem on a HW or exam and don't see why, attend office hours to ask about it. We are here to aid your learning process.

- Form good habits early on. We have a large amount of material to cover which means you will need to spend time outside of lecture reading and experimenting. Things will move **fast** so don't fall behind.

- There is a wealth of information on data structures online. Don't be afraid to consult online tutorials if a particular topic is troublesome for you. However, **do not** misuse to the Internet to obtain solutions that you don't understand. Follow the PRIME-DIRECTIVE at all times.