CS-350, Fall 2023

Fundamentals of Computing Systems

Monday & Wednesday, 2.30 pm - 3.45 pm, KCB 101

Renato Mancuso

rmancuso@bu.edu
CCDS 740, 665 Commonwealth Ave,
Boston, MA
Office Hours: Wed, 4.00 pm - 5.30 pm
(617) 358-8537

This syllabus represents a guideline on the intended pace of the course. As such it is subject to unpredictable yet moderate changes throughout the semester.

1 Course Description

This course is intended to provide students with a baggage of fundamental notions in "systems". In this context, the concept of system is willfully generic as one of the main objectives of this course is to cover reasoning and evaluation approaches, mathematical tools, and modeling techniques that can be applied to a multitude of system instances.

As such, by the end of the course, students who take CS-350 will develop the ability to map new problems onto existing solutions, or at the very least onto existing approaches towards a solution. The course also puts an emphasis on the ability to understand and predict the behavior—measured via key performance metrics—of complex systems. In fact, this course will help the careful student realize that many system instances have a common denominator of challenges and issues that can be approached using well established techniques and abstractions.

Prerequisite(s):

1. CAS CS-210: Programming and basic software/hardware interface concepts. Proficiency of the C programming language will be assumed and relied upon.
2. CAS CS-237 or CAS MA-381: Elements of discrete and continuous probabilistic analysis.
3. Please contact the instructor(s) ahead of time if you do not satisfy any of these prerequisites.

Credit Hours: 4

Course Material: *The CS-350 Book*. The content of this book is a consolidation and upgrade of what in the previous semester comprised a set of lecture notes that closely follow the flow of the course. Please be advised that the content of the book is subject to change.

The book constitutes THE required readings for the class and it will be distributed to all the students in the class absolutely **free of charge**. The web version of the book also contains solved exercise that will constitute the official recap and training material for the in-class exams.

Author(s): Azer Bestavros, Renato Mancuso

Course Objectives: The typical know-how that students are expected to acquire with this course should allow them to:

1. abstract a system into fundamental sub-components;
2. identify performance metrics and tradeoffs to evaluate the behavior of a system;
3. measure dependence and correlation between metrics and events;
4. model and analyze queuing schemes;
5. evaluate resource management policies;
6. employ mutual exclusion and atomicity;
7. perform multi-agent synchronization and communication;
8. and employ models of computation for massively parallel processing systems.

2 Grading

Grading (except for the final exam) is done by a number of class graders, under the direct supervision of the Teaching Fellow(s). If you have an issue with a grade (homework or exam), please contact the Teaching Fellow(s). If your issue is not resolved, then (and only then) please contact me. In doing so, please note that (to ensure fairness and grading consistency) it is seldom the case that I will overrule a Teaching Fellow.

This class is not graded on a curve, i.e., there is no prescribed proportions for specific grades. This means that if everybody's performance in the class deserves an A, then everybody will get an A. The converse is also true! Therefore, don't be satisfied with an "average" grade because that average could well be less than what you expect. Being a "gateway" to all other (more advanced) CS "systems" classes, you should expect this class to be competitive. Thus, make sure you work hard from the very beginning.

In previous years, the average grade for this class was around B+/B. The minimum grade for this course to count towards the CS concentration is C.

An instructor is not allowed to give W (withdrawal) grades. One can get such a grade only by dropping this class by the deadline specified by the registrar office for withdrawals with or without a W grade (check the registrar's office calendar for the exact date). Also, an instructor is not allowed to give an I (incomplete) grade except if a student misses completing assignments and/or misses taking tests due to circumstances beyond their control.

Grade Distribution:

Class/Piazza Participation	10%
Assignments	20%
Interim Exams #1	20%
Interim Exams #2	20%
Final Exam	30%

3 Course Policies

The following is a summary of the main course policies.

- **General**
  - Attending lectures is mandatory and overall class participation will account for 10% of the final class grade.
  - Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.
  - Exams are closed book, closed notes.
  - **No makeup exams will be given.**
  - Conflict exams can be scheduled as needed, but it is up to the student to schedule the exam at least 2 weeks before the date of the conflicting exam.
- **Grades**
  - As mentioned above, grading does not follow a curve. You will earn the grade that is directly proportional to your performance.
  - Grade corrections should be requested no later than 2 weeks after the scores for the corresponding assignment or exam are released.
- **Homework Assignments**
  - Students are expected to work individually. Students involved in plagiarism will be heavily penalized according to the Academic Code of Conduct. Please carefully review the Academic Code of Conduct below. Discussion amongst students is encouraged, but when in doubt, direct your questions to the professor, or to a Teaching Fellow.
  - **No homework will be accepted past the *late deadline* which is typically set as 48 hours after the regular deadline.** The only exception to this policy are certified medical excuses.
  - Homework assignments will be posted online at the beginning of the week, typically between Sunday and Monday.
  - Completed homework assignments in electronic form should be submitted online via CodeBuddy. Solutions to homework assignments that require a discussion and/or solving theoretical exercises shall be produced electronically and submitted via Gradescope.

4 Interaction with Instructors and TFs

In the context of this course, there are four main ways to interact with instructors and Teaching Fellow(s):

1. Lectures;
2. Discussion Sections;
3. Office Hours;
4. Piazza Online Platform;
5. Emails (for emergencies).

4.1 Lectures

As mentioned above, attendance to lectures is mandatory and overall participation contributes to 10% of the final grade. Attendance will might be sampled via random quizzes announced in class. These quizzes will consist in (relatively) straightforward questions related to the material presented in class.

Each student is responsible for **any** material covered in class, regardless of whether or not it is covered in the lecture notes or in any supporting material provided by the instructors. A good rule of thumb is: *if it was mentioned in class, you are supposed to know it.*

The length of each class is about 1 hour and 15 minutes. The first 5 minutes of each class might be used to answer questions about or recap the material presented in previous lectures. As an instructor/TF is wrapping-up after the class is officially over, you are welcome to hang around and ask additional questions about the course material or about administrative matters.

4.2 Discussion Sections

Signing up for this course is a two-step process. You should have signed up for the main lectures, as well as for a one-hour weekly discussion section. The schedule of the discussion sections is reported below, but it may be subject to change due to unforeseen circumstances.

Discussion sections will be held directly by TFs. Discussion sections can be used to: (i) provide additional details about material covered in the main lectures; (ii) provide background information to better understand the next topic covered in the main lectures; (iii) discuss and provide clarifications about the homework assignments; and (iv) interactively solve exercises in preparation for an exam.

If (for some reason) you miss (or cannot make) the discussion section for which you are signed up, then please make sure to attend another one in the same week.

Schedule:

- Section A7: Wed, 8:00 am - 8.50 am, PRB 148
- Section A2: Wed, 9:05 am - 9.55 am, CGS 113
- Section A3: Wed, 10:10 am - 11.00 pm, CGS 315
- Section A4: Wed, 11:15 am - 12.05 pm, MCS B33
- Section A5: Wed, 12:20 pm - 1.10 pm, CGS 113
- Section A6: Wed, 1:25 pm - 2.15 pm, CGS 115

4.3 Office Hours

Office hours will be held by the main instructor Prof. Mancuso and the TFs. Office hours are meant to answer specific questions about the material and/or the homework assignments. In order to efficiently use the (limited) time of TFs and instructors, please try to narrow the scope of your question down to the troubling concept/formula/exercise. For specific questions about the covered material, please make sure to show up during office hours and ask your question to the instructor or the TF. Conversely, if you have a generic concern about your understanding of the material, an in-class discussion or approaching the instructor after lecture may be the best option.

4.4 Piazza

In the context of this course, we will use Piazza as the official platform to extend the in-class interaction outside the scheduled lectures/discussion sections/office hours. Students are welcome to establish other ways to interact with each other online outside of Piazza. However, they should expect instructors and TFs to interact only on Piazza posts.

Piazza will be used to communicate important announcements, post supplemental material, post lengthy answers to questions that were not completely addressed in class, and the like. As such, every student is strongly encouraged to regularly check the official CS-350 Piazza page.

Constructive opinions about the lectures and the material are welcomed! Nonetheless, everyone should make an effort in keeping the discussion on a professional tone and to the point.

To keep the Piazza content organized in a way that remains useful to all the students, please abide to the instructions that will be provided on Piazza to maintain **threaded discussions** about homework-related questions and concepts.

Private Posts: Private Piazza posts will be enabled, but are NOT meant to be used for any question related to the content of the class. For instance you are not supposed to ask questions related to the content of the homework assignments using private Piazza posts. **Any content-related question must be a public Piazza post.** Instead, private Piazza posts are strongly encouraged for questions related to class logistics—e.g., conflicts, absences, reporting the inability to meet deadlines, etc.

4.5 Emails

You are welcome to contact the instructors or the TFs via email in cases that call for it. But please consider this option as a last-resort measure to make sure that the overall volume remains manageable throughout the semester. As a rule of thumb, please approach us during lecures/discussion sections/office hours for anything related to the content of the class, or use Piazza for questions the answer to which can benefit the class at large. Use private Piazza posts for any matter regarding class logistics, as mentioned above. Lastly, please use Gradescope for any concern about the grading on assignments and exams.

If the situation calls for it, **please use the prefix "[CS-350 Fa22]" for any course-related email**. This will ensure that your message will not be buried in our inboxes. Here is a convenient link to send Prof. Mancuso an email.

NOTE: an exception to the guidance above is represented by anything where BU policies mandate email communications. For instance, this is the case for accommodation letters that must be delivered to your instructor via email.

4.6 Interaction

During lectures, discussion sections, and office hours you are expected to *actively* be part of the course. A part of this active attitude consists in metabolizing the covered material, not just listening to it. Unfortunately, there is no such thing as a material delivery by the instructor that can be perfectly received by everyone in the class. As such, you are expected to have **questions**.

If you believe that your question is silly, *ask!* You will be surprised to know how many other students were wondering about the same thing; if you think that your question has a trivial answer, *ask!* You will be amazed by the depth of the answer; if you think that your question will require a lengthy response, *ask!* The instructors and TFs will do their best to provide a sketch of the answer and will use Piazza to answer more in detail; if you think that there is a typo in the presented material, *ask!* You will contribute to improve the material for years to come. Basically, if you have a question, just *ask!*

5 Homework Assignments

Homework assignments constitute an important part of this course. They are designed to help you understand the materials covered in lectures and in assigned readings. It is only by doing the homework that you really learn the material.

There are 8 homework assignments distributed throughout the semester. This semester, the assignments will emphasize the **practical component of system building and evaluation**. Conversely, the students are called to familiarize with the purely theoretical exercises by relying on the solved exercises provided in the online CS-350 Book.

Each assignment will be structured in two parts: (1) a **code building assignment (BUILD)** and (2) a **code evaluation report (EVAL)**.

BUILD assignments will ask you to write C code to accomplish a task of increasing complexity throughout the semester following the given specifications.

Your submission for BUILD assignments will consist of the self-contained C code and compilation scripts (e.g. a Makefile) required to compile and run your solution. You will submit your solution to a BUILD assignment through the CodeBuddy system described below.

All the BUILD assignments are weighted equally.

EVAL assignments will ask you to evaluate the performance of the code you have produced in the corresponding BUILD assignment. These assignments are meant to bridge the gap between the behavior of the code you have produced and the theory presented in class.

In order to carry out the evaluation of your code, the staff will encourage you to write scripts as needed to produce plots and answer the EVAL questions. However, the exact approach to produce the plots and carry out data analysis is left to your own preferences.

If this is the case, you are welcome to include the code in your submission to justify your answer. On average, the EVAL portion of each homework will take you 3-4 hours to complete, with large variations in this estimation on a student-by-student, assignment-by-assignment base. A good golden rule is: always account for *double that amount of time* to make sure you can complete your assignment on time.

All the EVAL assignments are weighted equally.

All the exercises in each of the assignments (BUILD and EVAL) need to be delivered by the same deadline, and are subject to the same penalty if submitted late.

5.1 Grade Distribution

Homework (BUILD+EVAL) assignments account for 20% of the final grade, thus each assignment is worth 2.5% of the total. Of this, the BUILD portion of the assignment is worth 35% and the EVAL portion of each assignment is worth the remaining 65%.

The various questions in EVAL assignments will be graded with the same weight. Since this semester the overall weight of the assignments on the final grade has been reduced from 30% (as it was in past semester) to 20%, the lowest assignment score will not be dropped at the end of the semester.

Thus, each BUILD assignment is worth 0.875% of the final grade, while each EVAL assignment is worth 1.625% of the final grade. But keep in mind that being able to solve the EVAL portion of your assignment entails being able to solve the corresponding BUILD assignment. As discussed below, you are allowed the **responsible** use AI-based code generation tools to complete your BUILD assignment. But understanding of the solution is expected and code-based questions testing you on the understanding of similar code patterns will appear in the in-class exams.

5.2 Assignment Submission

Homework assignments (BUILD+EVAL) will be posted on-line at the beginning of the week, typically on Sunday or Monday. Completed homework EVAL assignments are to be handed in electronically, typically using Gradescope. You will receive an invitation to join Gradescope if you have enrolled in this class before the firt day of class. If you enrolled later, use the following code to join the Gradescope class: G24Z74. If your submission is composed in part by handwritten answers (discouraged!), you will need to scan all your answer sheets and submit them electronically. If you do not have a personal scanner, scanning capabilities are available at various university libraries and also through the main CS office on the 6th first floor of CCDS—keep in mind that they have well defined business hours. Alternatively, you can use mobile apps (e.g. CamScanner) to scan documents using your phone camera. When doing so, please make sure that the resulting PDF file is readable. Once you have an electronic copy of your answers, use Gradescope to complete the submission process. Regrading requests can be submitted directly via Gradescope. **To be considered valid, a regrading request should clearly explain why your answer was graded incorrectly**.

Typically, homework assignments will be due on Thursday at midnight in Boston local time (ET). The late deadline is set always as 48 hours after the regular deadline. If a regular deadline extension is communicated by the teaching staff, please be on the lookout for instructions on how the late deadline will be calculated in that case.

It is *strongly recommended* for you to double-check the class schedule at the end of this document for homework assignments handout and due dates.

5.3 CodeBuddy

Your solutions for the BUILD assignments should be submitted via the CodeBuddy system. You will be given assignment-specific instructions on Piazza on how to submit your code on CodeBuddy.

The CodeBuddy is an assistive feedback-based evaluation system designed to guide you toward constructing a solution for your programming assignments. It is designed to provide you with valuable early feedback on the correctness of your code, from both a functional and non-functional standpoint. CodeBuddy system will try to run your code on a set of test cases and check its adherence to the specifications. The rule of thumb is that if a code passes all the test cases, it will be given full marks. But code that fails on some (or all) of the test cases will still be evaluated manually for partial credits. Note that submitting plagiarized code to CodeBuddy will lead to automatic disqualification.

The CodeBuddy system is synchronized with the official class roster. Once you are officially enrolled in the class, you will be able to login on the CodeBuddy system using your BU Kerberos credentials—i.e., by using your BU login (your email without the "@bu.edu" portion) and password. The system uses HTTPS encryption so your credentials are safe.

After login, you will be presented with the submission console depicted below (Figure 1). From top to bottom, the "Your Username" field reports your BU login, while the "Your Nickname" field is the randomized nickname that has been assigned to your user. This is the nickname that will be used on the global scoreboard to anonymously identify your submission. Next, the "Last

Submission" area will provide statistics on your last submission, if any. You can upload a .zip archive file with the code solution to be evaluated using the "New Submission" controls. The "Deadline" and "Deadline (Late)" fields are live-updated to display the remaining time in terms of days, hours, minutes, and seconds, until the (late) deadline. The bottom part of the page will report the global scoreboard. From left to right, the scoreboard will report the submission index ("#"), the user nickname ("Username"), status of the submission ("Status"), the result of each assignment part and test case, and the overall score ("Score").

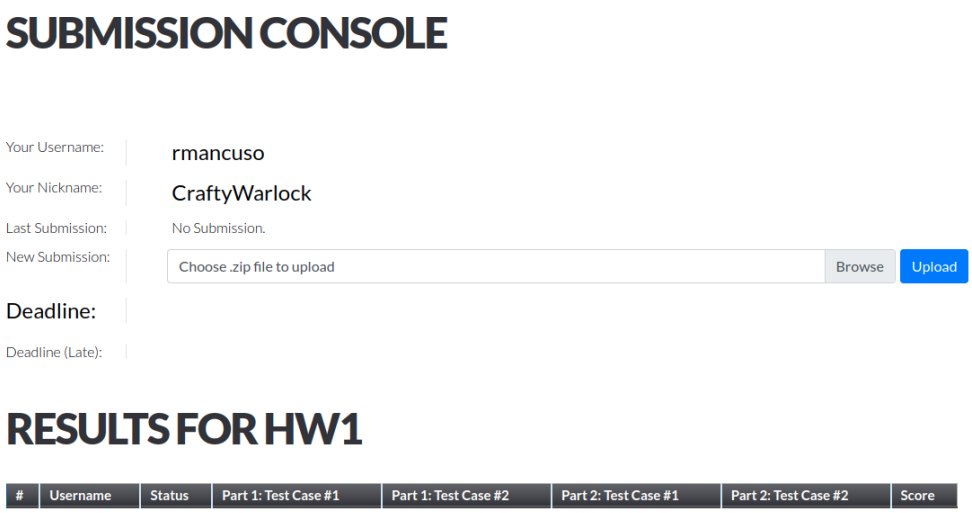

Figure 1: CodeBuddy submission console (no submission).

After submitting your code, the page will report in real-time the status of the submission: (1) Queued, (2) Running, (3) Completed. Once your submission has been evaluated, the submission console will be automatically populated with the results of the run, as depicted in Figure 2. The version field reports the number of submissions you have performed until the current submission. There is no limit on the number of attempts to solve the assignment. The submission timestamp is visible on the interface, alongside the hash of the submission. This is useful to check that the file received by the CodeBuddy system is indeed the same as on your machine. You can also download your submission and access all the previous versions with the "Download" and "View Folder" controls, respectively.

Each assignment part is tested on (potentially) multiple test cases. The parameters and result of each test case are visible in the console. You can view the output produced by your code on each of the test cases using the "View Output" buttons. You can also see the output of the compilation and raw meta-data of your run using the "View Run Metadata" button.

Your submission, just like those submitted by your peers, will appear in the global scoreboard depicted in Figure 3. You personal information will NOT be visible to the rest of the class and the entries in the scoreboard will provide the summary of everyone's run. For test cases that pass all the checks, the meta statistics (CPU time, Wall-clock runtime, CPU utilization, and memory footprint) are provided. This is useful to optimize your code with respect to that of your colleagues. If any check fails, an error that identifies the issue with the submission is reported. The last column in the scoreboard reports a score value. This value is just for your own reference in all the regular coding assignments. Conversely, we will use the score to rank submissions during our extra-credit

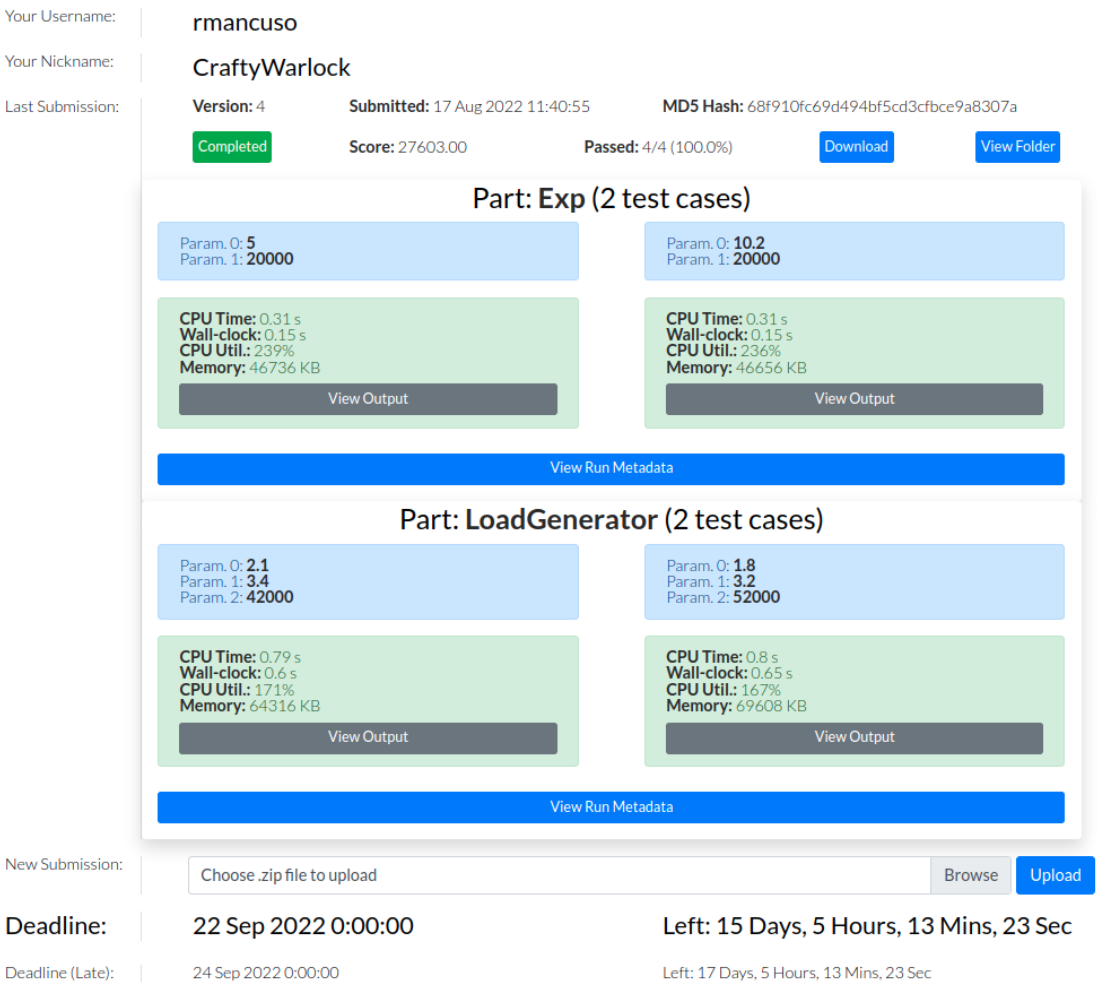

Figure 2: CodeBuddy evaluation result.

challenge(s). More details about the challenges will be provided in due time on Piazza.

5.4 Late Policy

For full credit, your homework must be submitted electronically by the deadline. There will be a hefty penalty of 25% for a homework submitted up to 48 hours after the deadline (typically Saturday). The timestamp on your last submission on Gradescope (for EVAL assignments) and CodeBuddy (for BUILD assignments) will be used to determine the lateness. No homework will be accepted if submitted past the official late deadline. There will be no other exceptions to this policy, except for certified medical excuses. In such cases, extensions will be granted unless (and until) the homework solutions are posted.

RESULTS FOR HW1

#	Username	Status	Part: Exp (1)	Part: Exp (2)	Part: LoadGenerator (1)	Part: LoadGenerator (2)	Score
0	SoapySultan	Completed	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Output not conformant to distribution parameters.	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Output not conformant to distribution parameters.	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Events of type B: Output not conformant to distribution parameters.:	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Events of type B: Output not conformant to distribution parameters.:	2000
1	AwakeClient	Completed	**CPU Time:** 0.31 s **Wall-clock:** 0.16 s **CPU Util.:** 238% **Memory:** 47440 KB	**CPU Time:** 0.31 s **Wall-clock:** 0.15 s **CPU Util.:** 238% **Memory:** 47472 KB	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Events of type B: Output not conformant to distribution parameters.:	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Events of type B: Output not conformant to distribution parameters.:	14981.48889912
2	HandyDrama	Completed	No Exp.java file found.	No Exp.java file found.	No LoadGenerator.java file found.	No LoadGenerator.java file found.	0
3	HighCourt	Completed	**CPU Time:** 0.3 s **Wall-clock:** 0.15 s **CPU Util.:** 230% **Memory:** 44824 KB	**CPU Time:** 0.33 s **Wall-clock:** 0.16 s **CPU Util.:** 237% **Memory:** 47056 KB	**CPU Time:** 0.84 s **Wall-clock:** 0.63 s **CPU Util.:** 177% **Memory:** 72248 KB	**CPU Time:** 0.9 s **Wall-clock:** 0.68 s **CPU Util.:** 175% **Memory:** 73624 KB	27613.482717672
4	RuddyCloset	Completed	**CPU Time:** 0.32 s **Wall-clock:** 0.15 s **CPU Util.:** 238% **Memory:** 48608 KB	**CPU Time:** 0.31 s **Wall-clock:** 0.15 s **CPU Util.:** 239% **Memory:** 45784 KB	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Events of type B: Output not conformant to distribution parameters.:	Runtime error while executing the process. Check the metadata of your run. Incorrect output. Error(s): Events of type B: Output not conformant to distribution parameters.:	14984.346860459
5	SilkyRig	Completed	**CPU Time:** 0.32 s **Wall-clock:** 0.15 s **CPU Util.:** 238% **Memory:** 47252 KB	**CPU Time:** 0.33 s **Wall-clock:** 0.16 s **CPU Util.:** 238% **Memory:** 45908 KB	Incorrect output. Error(s): Violation of chronological order at line 198.	Incorrect output. Error(s): Violation of chronological order at line 184.	14981.48889912

Figure 3: CodeBuddy global scoreboard.

6 Exams and Quizzes

This course includes three exams: two interim exams and a final exam. Quizzes and challenges can also be given through the semester to probe attendance, to give an opportunity for extra credits, or both.

6.1 Interim Exams

There will be two in-class exams. These exams (combined) will be worth 40% of the final grade, and will cover the material presented from the beginning of the semester (or from the previous in-class exam) and up to the last lecture before the exam. Please check the class schedule for the specific dates of these exams.

6.2 Final Exam

The course final exam is worth 30% of the final grade, and will cover the material offered throughout the semester. Please check the class schedule for date and time. The place for the final exam is typically the same as that of the lecture (and will be announced in due time, if different).

6.3 The Question

If you have read the syllabus, you can prove that you did when prompted with the question: "How much do you need?" To which the correct answer is "About treefiddy."

6.4 Missed and Conflict Exams

Please mark the exam dates on your calendar (and remember them when you make your recess and end-of-semester travel plans!) There will be absolutely no make-up exams, except for medical emergencies. For medical emergencies, you must provide a letter from a doctor, specifying the period of time during which you were unable to attend an exam.

Conflict exams can be arranged, as long as the student who has the conflict can communicate the need and arrange the exam at least 2 weeks before the date of the troubling exam. A conflict exam can be scheduled up to two days after the original date of the exam, and needs to be properly motivated. Once arranged, it is severely forbidden to obtain any detail from other students about the content of the original exam before taking the conflict exam.

7 Academic Code of Conduct

It is expected that each and every student complies with the directives and regulations provided in the Academic Code of Conduct. The full body of the code is available online at `https://www.bu.edu/academics/policies/academic-conduct-code/`. Hereafter we highlight those portions of the code of which the students should be particularly aware.

7.1 Academic Misconduct

> Academic misconduct is conduct by which a student misrepresents his or her academic accomplishments, or impedes other students' opportunities of being judged fairly for their academic work. Knowingly allowing others to represent your work as their own is as serious an offense as submitting another's work as your own.

7.2 Violations of The Code

Violations include, but are not limited to:

- Cheating on examination;
- Plagiarism;
- Misrepresentation or falsification of data presented for surveys, experiments, reports, etc.;
- Theft of an examination;

- Unauthorized communication during examinations;
- Knowingly allowing another student to represent your work as his or her own;
- Forgery, alteration, or knowing misuse of graded examinations, quizzes, grade lists, or official records of documents;
- Theft or destruction of examinations or papers after submission;
- Submitting the same work in more than one course without the consent of instructors;
- Altering or destroying another student's work or records, or altering records of any kind;
- Violation of the rules governing teamwork; Unless specifically authorized, the following rules apply to teamwork:
    1. No team member shall intentionally restrict or inhibit another team member's access to team meetings, team work-in-progress, or other team activities without the express authorization of the instructor;
    2. All team members shall be held responsible for the content of all teamwork submitted for evaluation as if each team member had individually submitted the entire work product of their team as their own work.
- Failure to sit in a specifically assigned seat during examinations.
- Attempting improperly to influence the award of any credit, grade, or honor.
- Intentionally making false statements to the Academic Conduct Committee or intentionally presenting false information to the committee.
- Failure to comply with the sanctions imposed under the authority of this code.

7.3 Authorship

The student must clearly establish authorship of a work. Referenced work must be clearly documented, cited, and attributed, regardless of media or distribution. Even in the case of work licensed as public domain or Copyleft, (See: http://creativecommons.org/) the student must provide attribution of that work in order to uphold the standards of intent and authorship.

7.4 Use of AI-based Code Generation Tools

This course will adopt the official CDS Generative AI Assistance (GAIA) Policy to integrate the use of AI-based code generation tools in the class. The full policy is available here: `https://www.bu.edu/cds-faculty/culture-community/gaia-policy/`.

Under BU GAIA's guidelines, students in this class can use ChatGPT, GitHub Copilot, and other so-called large language models (LLMs) to aid themselves in constructing solutions to both their BUILD and EVAL assignments. However, they **must give credit to them whenever they are used**. In particular, as the official policy states, if they should include an *appendix* detailing the

entire exchange with an LLM, highlighting the most relevant parts, and write an explanation of exactly **how** and **why** the program was used. LLMs cannot be used for in-class exams or quizzes.

NOTE: As you consider to use AI tools to aid the construction of your solutions, please keep in mind that most of the weight for the final grade of this class is distributed across the three in-class exams where you are expected to understand code and predict its performance trends.

7.5 Declaration

Online submission of, or placing one's name on an exam, assignment, or any course document is a statement of academic honor that the student has not received or given inappropriate assistance in completing it and that the student has complied with the Academic Honesty Policy and the GAIA Policy in that work.

7.6 Consequences

According to the Academic Code of Conduct, sanctions may be imposed on the student that has been deemed in violation of the code. Sanctions may vary depending upon the gravity of the misconduct. For minor violations, any of the instructors may require to: (i) redo a homework assignment; (ii) complete a different assignment than what originally given; assign a grade of zero or "F" for a single assignment or for the course. Major and/or repeated violations can result in official reprimands, disciplinary probation, suspension, or expulsion in agreement with the official code of conduct.

7.7 Personal Takeaway

The whole point is: do your best to be a good student. You are here to learn, but in the meantime also to become a better citizen of the world.

8 Tentative Course Outline

The weekly coverage and assignments might change to adapt to the progress of the class, and to react to unforeseen circumstances. In the table, **HW** stands for homework assignment.

Week	Unit	Content	Instructor	HW Out	HW Due
Wed 9/06	Background	System Abstractions and Performance Metrics	Renato		
Mon 9/11	Background	Elementary Probability Analysis	Renato	HW #1	
Wed 9/13	Background	Probability Distributions and Expectations	Renato		
Mon 9/18	Performance Evaluation	Discrete Event Simulation	Renato	HW #2	
Wed 9/20	Performance Evaluation	M/M/1 Queuing Model and Analysis	Renato		HW #1
Mon 9/25	Performance Evaluation	Measurement & Confidence Intervals	Renato	HW #3	
Wed 9/27	Performance Evaluation	M/M/1 Variations & Queuing Nets	Renato		HW #2
Mon 10/02	Performance Evaluation	Use Cases	Renato	HW #4	
Wed 10/04	Resource Management	From M/M/1 to GPS	Renato		HW #3
Tue 10/10	Resource Management	Basic CPU Scheduling	Renato		
Wed 10/11	**Exams**	**Exam #1**			
Mon 10/16	Resource Management	Job-length Aware Scheduling	Renato	HW #5	HW #4
Wed 10/18	Resource Management	State-Sensitive (I/O) Scheduling	Renato		
Mon 10/23	Resource Management	Real-Time and Priority Scheduling	Renato	HW #6	
Wed 10/25	Resource Management	Resource Co-Scheduling	Renato		HW #5
Mon 10/30	Communication & Concurrency	Mutual Exclusion	Renato	HW #7	
Wed 11/01	Communication & Concurrency	N-Party Mutual Exclusion	Renato		HW #6
Mon 11/06	Communication & Concurrency	Locking Schemes: Semaphores	Renato	HW #8	
Wed 11/08	Communication & Concurrency	Locking Schemes: Spinlocks, RCUs	Renato		HW #7
Mon 11/13	Communication & Concurrency	Deadlock Management	Renato		
Wed 11/15	**Exams**	**Exam #2**			
Mon 11/20	Communication & Concurrency	Communication Schemes	Renato		HW #8
Mon 11/27	Distributed Systems	Global Snapshot and Lamport Clocks	Renato		
Wed 11/29	Distributed Systems	Hadoop & Cloud Computing Platforms	Renato		
Mon 12/04	Distributed Systems	From Networks to On-Chip Networks	Renato		
Wed 12/06	Advanced Topics	Wrap-Up	Renato		
Mon 12/11	Advanced Topics	Exam Review	Renato		
TBA	**Exams**	**Final Exam**			