Syllabus – Winter term, 2020-21

This document contains the following sections. Click on a section to go directly to that section of the document, or just read the document from top to bottom (your choice).

Catalog Description

CSSE 120 Introduction to Software Development
3R-3L-4C F,W,S

An introduction to procedural and object-oriented programming with an emphasis on problem solving. Problems may include visualizing scientific or commercial data, interfacing with external hardware such as robots, or solving numeric problems from a variety of engineering disciplines. Procedural programming concepts covered include data types, variables, control structures, arrays, and data I/O. Object-oriented programming concepts covered include object creation and use, object interaction, and the design of simple classes. Software engineering concepts covered include testing, incremental development, understanding requirements, and teamwork.

Prerequisite: None.

This course does *not* assume that you have any prior programming experience. But, like many other Rose-Hulman courses, CSSE 120 requires that you keep up with each day's work. By the end of the first week, you *will* have a quite a bit of software development experience, or you will be behind!

Instructors

You are always welcome to contact any of us — **we work as a team.**

Sana Ebrahimi (Section 4) ebrahimi@rose-hulman.edu	Sriram Mohan (Section 1) mohan@rose-hulman.edu	David Mutchler (Sections 2 and 3) mutchler@rose-hulman.edu
		

Academic Integrity

Maintaining your academic integrity is an important part of your life and is ALWAYS better than not doing so, in the long run (and probably the short run as well).

If you are tempted to cheat, don't. Come talk to us instead.

Both receiving AND GIVING help inappropriately are **violations** of academic integrity.

The Student Handbook includes procedures regarding academic misconduct and includes:

> "Rose-Hulman expects its students to be responsible adults and to behave at all times with honor and integrity."

In particular, for CSSE 120:

Exams are to be done INDIVIDUALLY, with NO COMMUNICATION with anyone other than your instructor and their delegates. Each exam will have explicit rules for what external resources you may use for what parts of the exam.

For ***all work other than exams***, getting help and working with others is encouraged. Just follow these three rules for all work that you turn in as your own:

- **Rule 1: Attempt the problem yourself first** (or with your group in a group exercise).

  It is NEVER right to begin by copying someone else's solution.

  If you find a complete solution on the internet to the very same exercise that you are doing, you may NOT use that solution in any way. Just ignore it!

- **Rule 2: Make a sincere effort to *understand* the concepts behind the problem on which you are getting help.**

  If you do NOT understand your own code, it is NOT yet your own work; instead, ask your instructor or a course assistant for help!

- **Rule 3:** Unless specified otherwise (as on exams), **you may use the following resources:**
  - ***Any materials that your instructor supplied.***

    For example, you may copy an example from code that we supplied when you are working on your own code, modifying the example as needed.

    Warning: Be cautious about doing such copying too much. There is a lot to be said for typing (rather than copying) your code, to help develop "muscle memory".

  - ***Any explanations (but NOT examples of code) from any source, including humans.***

    Warning: Be cautious about getting help from anyone other than your instructors and the CSSE 120 student assistants. They are specially trained to help you ***learn***, not merely solve the problem.

  - ***Small snippets of code (say, fewer than 10 or so lines of code) from any source.***

    For example, snippets from the excellent ***Stack Overflow***.

    But recall Rule 2 above. Do not use a snippet of code if you do not understand it!

The normal **penalty** in this course for ***getting or giving*** help to another student on an exam is an F in the course. Repeated Academic Misconduct will lead to the student being dismissed from Rose-Hulman.

If you are tempted to cheat, don't. Come talk to us instead.

Assessment (how you earn your grade):

Rule 1: To pass the course, you must complete (to the satisfaction of your instructor) **ALL the *Preparations* and ALL the *PyCharm coding* exercises.**

Note: You may redo any quiz or exercise until it is satisfactory, if you do so in a timely way. (See Attendance / Late work.)

Rule 2: Assuming that you succeed per Rule 1, calculate the weighted average of your scores on exams and your capstone team project, and then convert that weighted average to a letter grade, as follows:

Exam 1: 10%	**Exam 2: 20%**	**Exam 3: 35%**	**Capstone Project: 35%**

There will also be an *optional* Final Exam, with its score averaged with Exam 3 (together counting 35% of the total grade). We reserve the right to change that Final Exam to *required*, if announced just after Exam 3 or earlier.

Convert from a percentage to a letter grade by:

A:	B+:	B:	C+:	C:	D:	F:
92 and up	88 to 92	83 to 88	78 to 83	72 to 78	67 to 72	Below 67

Note that this is NOT the standard "10 point" scale.

Rule 3: Additionally, to earn a C or better in the course, you must achieve ALL of the following:

- At least a C average on the paper-and-pencil exams, and
- At least a C average on the on-the-computer exams, and
- At least a C average on the capstone team project.

Special needs (and how we accommodate them):

Rose-Hulman, and the instructors of this course in particular, are committed to working with students who have special needs or disabilities.

We understand that "invisible" disabilities (learning and attention deficit disorders, chronic fatigue syndrome, depression, anxiety, etc.) can significantly affect a student's academic performance.

We strongly encourage students to document special academic circumstances with the staff at the Student Accessibily Services, and then to contact us as soon as possible so that we can work together to provide recommended academic accommodations while protecting your privacy. Please note that it is the student's responsibility to request any approved, documented academic accommodations (such as extra time) at least one week in advance of exams.

Always **feel free to let us know about anything** that is causing your academic performance to be less than you wish it were. **We want to help you succeed in our class.**

Health-related resources **available to students for free** include:

Health Services Student Counseling Center (SCC).

From SCC's website: Examples of concerns typically addressed through our center include:

- Mild to moderate mood concerns including depression, anxiety, self-esteem, social anxiety, etc.
- Adjustment and developmental concerns such as transitioning to college, new life circumstances, cultural adjustment, homesickness, etc.
- Interpersonal concerns including adjusting to living with roommates, conflict with friends and/or family, adjusting to relationship difficulties including break-ups, communicating thoughts and feelings to others, being assertive, etc.
- Identity development including various dimensions such as sexual, racial/ethnic, religious, gender, etc.
- Academic related concerns impacted by difficulties with sleep adjustment, time-management and self-discipline, low motivation, perfectionism, interpersonal communication, etc.
- Grief and loss.
- Recent trauma including sexual assault, physical assault, injury, etc.

To make an appointment, call **812-877-8537**. We are located in the northeast corner on the upper level of the Mussalem Union. We are open **Monday-Friday 8AM-5PM.**

Making our classroom welcoming

You can do your part to ensure a *welcoming, professional classroom climate* by:

- **Be supportive** of your classmates. Offer help where appropriate. Help your classmates feel confident and welcomed.
- **Project respect to classmates**, course assistants and instructors. (And tell us, as described below, if you feel anyone is being disrespectful of you or others in the classroom.)
- Speak to your classmates, course assistants and instructors with **courtesy and professionalism**. The classroom atmosphere is informal, but avoid off-color jokes, rude language, and anything non-course-related that might offend classmates, your instructor, course assistants or visitors.

  Similarly, be sure that your computer screen shows only courteous and professional images and words, at all times during class.

  If you wouldn't say or show it in front of your grandmother, don't say it in our class!

  Avoid words and behaviors that might be perceived as confrontational or aggressive. Strive to **avoid negative "you" phrases** like "you are wrong" or "you need to ...". **Instead**:
- **Use neutral "I/us" phrases** like "I think that that approach is not right" or "Perhaps it would be helpful for us to …" or "I wonder if ..."
- **Don't do anything that will detract from your** learning or that of people around you. Such things include failing to set your phone to "silent" mode, having loud notifications going off on your computer, talking loudly, chewing gum noisily, not taking adequate care of your personal hygiene, and arriving late for class.
- **Restrict all your conversations in class to things related to the class**, except for times when your instructor allows you to do otherwise (as in a between-periods break, for example).

If you are experiencing issues that may make you less able to be respectful to others in the classroom, consider alerting your instructor to that fact. Additionally, we strongly encourage any student who is feeling stress or experiencing any sort of difficult issue to Student Counseling Center. They are an excellent resource available to students for free.

If at any point, you are not comfortable in the classroom, for ANY reason, or you observe any behaviors by ANYONE (classmates, course assistants, your instructor or visitors) that may make the classroom climate feel less welcoming for students, ***please tell us*****.** Ways to do so include:

- Talk to your instructor privately outside of class.
- Talk with any professor or staff member with whom you feel comfortable, perhaps one of the following CSSE professors are among those who agreed to be an ombudsman to help ensure that our classroom climates are welcoming to all: Professors Amanda

Stouder (stouder@rose-hulman.edu), Kim Tracy (tracy@rose-hulman.edu), and Robert Williamson (williarj@rose-hulman.edu). Ask them to act on your behalf in ways with which you are comfortable.

- All of your professors will do their utmost to ensure confidentiality, but sometimes you might feel the need to say something anonymously. You can do so going to Moodle for our class, and noting the *Expressing Concerns Anonymously* tool near the top. All we ask is that you not use it just to vent when you are momentarily frustrated. Take a deep breath, count to 10, and if you still want to express a concern anonymously, do so.
- Or, if you want to express concerns in a completely confidential way, the Student Counseling Center is an excellent resource.

Attendance and Late Work:

For sections that meet synchronously (i.e., at a scheduled time), we expect you to attend unless you are ill or have arranged to miss class (for a job interview, sporting or club activity, or important personal activity like a wedding, for example). Attendance during the Capstone Project is especially important since you interact with your team at those sessions.

Re Late Work: Our class meets for 30 sessions during the term. When you arrive to class (possibly via remote communication) for Session N, you should normally have completed all the work for Session N-1 plus the *Preparation* for Session N, except as follows:

- If there is any work about which you are confused in the *Coding* for Session N-1 or the *Preparation* for Session N, try to seek help outside of class, but you may always bring such questions to Session N.

If you fall behind by more one full session (e.g., you enter class for Session 5 and have not yet started Session 4), get help ASAP from your instructor to help you catch up and to set appropriate individual deadlines for you doing so. As long as you do complete the work by whatever day you and your instructor agree upon, you get full credit for it.

Student outcomes (learning objectives):

Students who successfully complete this course should be able to:

Analyze, explain and use appropriately in coding: ***Fundamental programming concepts*** **including:**

1. Syntax and semantics
2. Objects, types, names (variables), expressions, and assignment
3. Branching control structures
4. Explicit loops, both definite and indefinite
5. Functions, parameter passing, user-defined functions

6. Constructing objects, and using their methods and instance variables (fields)
7. Components of a class, as expressed in code as well as in Unified Modeling Language (UML) or other such diagrams
8. Sequences, including lists and strings
9. Indirection, box-and-pointer diagrams and mutable objects
10. Input and output, to both consoles and text files
11. Modularity and structured decomposition to break a program into smaller pieces
12. Using an application programming interface (API)

Design, implement, debug and test small programs for solving problems motivated by real-world interests, using the *above concepts* and ***modern software engineering practices*** including (where appropriate, and at an elementary level):

1. An appropriate integrated development environment with version control
2. Coding to a specification
3. Iterative enhancement
4. Pair programming
5. Test-first programming
6. Documenting software, for internal readers and for external readers
7. Use of application programming interfaces (APIs)

Work for 2 - 4 weeks in a team of 3-4 students on a small software development project, demonstrating (at an elementary level) effective use of:

1. Division of labor
2. Integrating teammates' work
3. Modularity and structured decomposition to break a program into smaller pieces
4. Constructing objects from new APIs as needed, and using their methods and instance variables (fields)
5. Agile software development processes
6. Team roles
7. Conflict resolution