

## How to Succeed in CMPU 101 by Really Trying

I want you to be successful in CMPU 101 this semester. “Success” in the course is more than just good grades. It means that you are being challenged to grow as a learner, that you are engaging actively with tasks that feed your growth, and that you are creating excellent work in computer science by completing tasks with an appropriate level of support. It also means that you are building your lifelong learning skills so that once the course is over, you are better and stronger as a learner, both so you can succeed in courses that have CMPU 101 as a prerequisite – and so you can continue to learn new things independently.

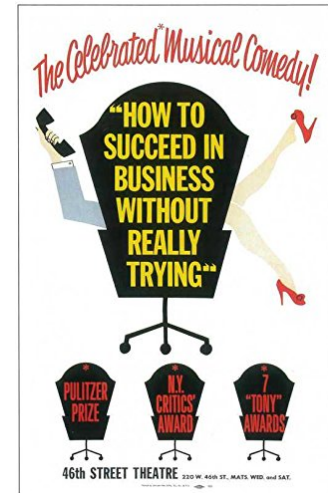
This is designated as a quantitative course, satisfying one of Vassar’s requirements for graduation. Like any quantitative course, it will challenge you to think abstractly, analytically, and logically. It’s my sincere hope that you enjoy this class, but the unavoidable truth is that learning computer science takes time, effort, and practice. So, here’s some essential advice:

### View CS as learning processes, not facts

In some courses, you basically need to learn a set of facts and formulas and when to apply them. Programming is different: much of it is about learning a process of creating something or solving a problem, not about memorization. Don’t expect to learn the material by reading it over and over. Instead, work on learning the steps we take and the questions we ask to make progress on problems.

### Take notes in class

While the slides and code from each class are posted to the course website for your reference, it still helps to take notes. Rather than write down what’s on the slides or try to write down everything I say, put the concepts into your own words. You may find it especially helpful to draw diagrams showing how different ideas relate.



*Close – but try harder!*

You should also pay attention to the questions other students ask and the answers to them. These won't be in the slides or code we release, but they may be the same questions you're wondering!

## Stay on top of the material

The course starts out with basic concepts, but each week builds on what was seen previously. Missing one class may prevent you from fully understanding the next, which may prevent you from completing a lab or assignment, and so on.

If there's something you don't understand, you need to make an effort to clarify it when it comes up.

I recommend you read the required material after the class where it's introduced. This way, you are reinforcing what you already understand with another explanation.

Focus on the main concepts and not just the details. For example, don't memorize each program you're shown, but try to understand why it's written the way it is – how is the whole put together?

It's tempting to focus all your time on the current assignment, but previous misunderstandings will come back to haunt you! When your work is graded, *go over it again*. What should you do differently when you see a similar problem in the future? What did you misunderstand?

## Start assignments early

If you leave assignments until the night before, you'll struggle just to finish before the deadline, which means you'll be working harder and learning less!

Start the assignments several days before they are due. The problems really benefit from going away and coming back later when you are stuck. And if you start early, you'll have time to get help if you realize there's something you don't understand.

## Know when to struggle and when to stop

It's important to make a serious effort to understand the material and to work through the assigned problems. Always asking for help

may let you complete your assignments, but it's a terrible way to learn from them – and it sets you up for failure on the exams!

On the other hand, there's no prize for spending hours and hours hitting your head against a problem. If you've made a sincere effort to answer an exercise, reviewed the lectures and textbook, and you're still not sure how to proceed, ask for help!

An important part of growing as a learner is figuring out the right amount of perseverance – when to keep struggling and when to get help.

## Learn together

Computer scientists have an undeserved reputation for being loners. In fact, computer science is a highly collaborative field, and computer scientists love to explore new ideas together. Remember that *you're not alone in this course*.

If you're having trouble understanding an idea, you should talk with your professor, coaches, and fellow students. It's a good idea to study together, since explaining concepts to peers and hearing their explanations can help to clarify your understanding.

## You can do this!

People often believe that programming and computer science is something that some people are “born” to do while others are not. Don't fall into this trap! Instead, approach CS as something that must be learned with practice. Some of your classmates have programmed before. Unsurprisingly, that means they are better at it than you are right now, but that doesn't mean that you can't learn to do this too!

You'll find `CMPU 101` to be very enjoyable if you embrace the challenge. Although hard work doesn't always feel good in the moment, by committing to learning, I think you will find that growth is fun and computer science is really interesting!

Sarah Andersen



© Sarah Andersen

### *Acknowledgments*

This guide is based on documents by Kathi Fisler (Brown University), Nancy Ide (Vassar College), Keith Schwarz (Stanford University), Marc Smith (Vassar College), and Robert Talbert (GVSU).