

Instructor – Michael Trigoboff

My office is on the [Sylvania Campus](#) in TCB 312. If you need office time with me you should communicate with me about when we can do that.

If you need to communicate with me privately, you should use PCC email. My email address is **michael.trigoboff@pcc.edu** Desire2Learn is now using PCC's GMail service for email, so you can send messages via MyPCC or Desire2Learn. Either way is OK.

This course is often taught as parallel sections, one in the classroom and the other via distance learning. When that is the case, distance students who want to attend classroom sessions are always welcome.

If you're interested, you can find out more about me at <http://spot.pcc.edu/~mtrigobo>.

Course Description

This course will introduce you to what's "beneath the surface" of higher-level programming languages like C and C++, namely the "object code." You will learn things about object code that will make you a more capable programmer. There's no substitute for knowing what's really going on "under the hood," and this course will teach you about that.

In this course, we will mainly be working with Unix. Unix skills are valuable and acquiring them should increase your employment opportunities. If a job interviewer asks you to demonstrate some Unix skills, you don't want to be sitting in front of that workstation wondering, "What do I do now?"

Unix is available to you on the department's Linux machine, **syccuxas01.pcc.edu**. You will be required to use this machine for the programming assignments.

You will be using the **C** programming language in this course. All of the programming assignments should be done in C within a Unix environment.

In addition to acquiring skill with Unix, you should also develop proficiency with one of the source code text editors that are common in Unix. Your choices are:

- **vim**
 - the power you need for programming
- **emacs**

- much more power than you need, has a steeper learning curve than **vim** (emacs is actually the programming language Lisp masquerading as a text editor)

I use **vim**, but you are free to choose **emacs** if you want something much more powerful. If you're at that job interview and they find out that you can't do text editing in Unix, they will *not* be impressed.

Learning Outcomes

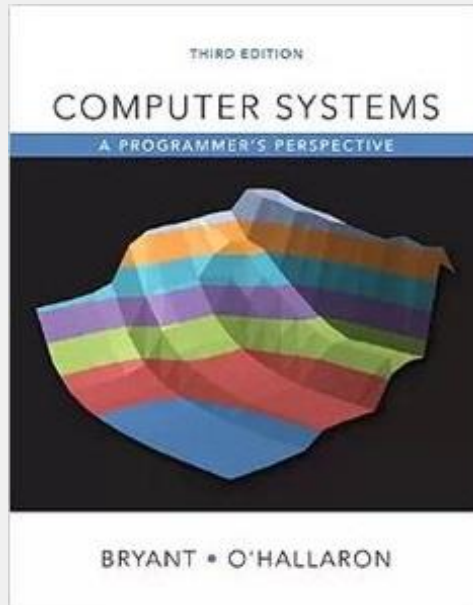
In this course, you will learn about the following topics:

- how to do programming using Unix and the C language
- what's "under the surface" of your C/C++ code
 - object code (assembly code, machine language)
 - the machine itself (which will be studied more extensively in CS201)
 - what the machine is actually doing when it executes your code
- deeper knowledge of how your programs work
 - special features of the Unix operating system
 - performance optimization
 - understanding of the true nature of the underlying Intel IA-32 architecture
- Intel's IA-32 Architecture
 - the Intel Assembly Language instructions commonly seen in object code
 - the machinery underlying the Intel IA-32 architecture
 - for instance, registers like **eax** are merely a convenient fiction
- the structure and function of "the stack" (also known as "the calling stack")
 - detailed understanding of "function calling overhead"
 - understanding of the nature of "buffer overflow attacks"
- exceptional control flow
 - functions that return twice, functions that never return, etc.
- techniques for optimizing your code to make it run faster
- opportunity to practice and improve your programming skills

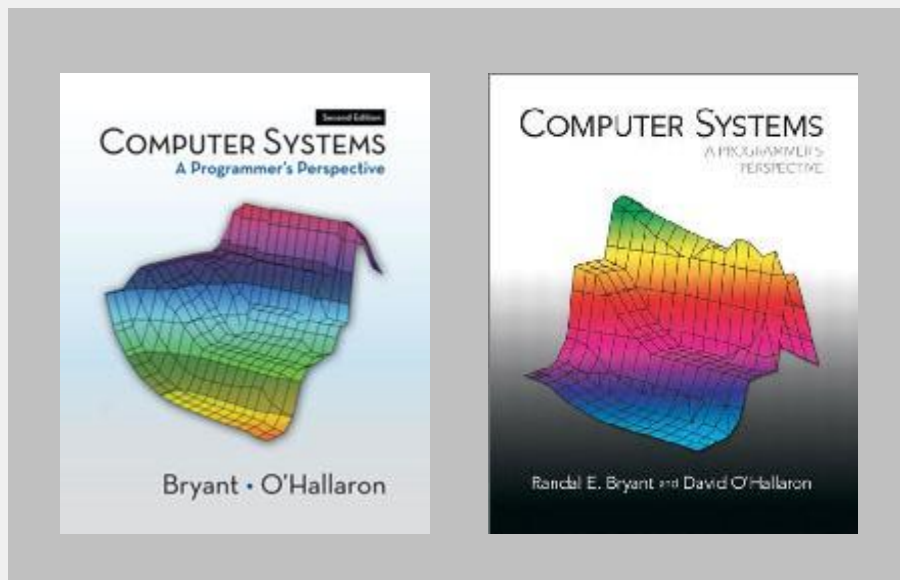
For more information, please go to the Course Content and Outcome Guide for this course at: www.pcc.edu/ccog.

Textbooks

The textbook for this course is [Computer Systems: A Programmer's Perspective](#) by Randal E. Bryant and David R. O'Hallaron. It is available at the [PCC bookstore](#).



You can get a used copy of the [second edition](#), or the first edition instead, and save yourself some money.

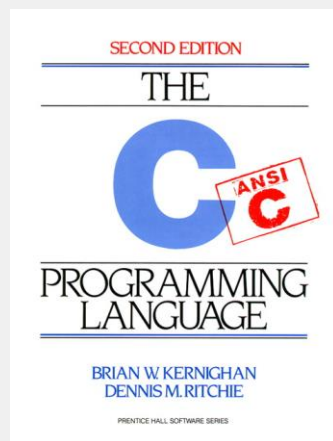


Make sure to go to the textbook site's errata page ([third edition](#), [second edition](#)). You will save yourself *hours* of futile effort attempting to understand incorrect (and therefore incomprehensible) material.

The errata page for the first edition is no longer available on the textbook's website, and it had *a lot* of errors. Fortunately, the errata page for the first edition is still available via [the Wayback Machine](#).

If you need a reference to the C programming language, there are many available online. The [GNU C Tutorial](#), for instance is usable, but occasionally it expresses strong opinions that I don't necessarily agree with (e.g. regarding the goto statement).

If you want a book, [The C Programming Language, 2nd Edition](#) by Brian W. Kernighan and Dennis Ritchie is a classic of Computer Science. Kernighan and Ritchie created the C language.



Course Work

The work for this course will consist of **online quizzes** and **programming assignments**. The quizzes are available in the **Quizzes** section of the course. The assignments are available in the **Assignments** section of the course. The assignments will sometimes (but not always) consist of a **Regular** part and a **Premium** part (in this case, you only need to do *one* of the parts).

How To Submit Programming Assignments

Personalize Your Code

Your code should always print out the following as its first line of output:

CS201 - **assignment** - **your name**

Make sure you change the places marked in **bold text** to show me which assignment you are submitting and who you are.

For example, if I were submitting assignment 3, my code would print out:

CS201 - A03 - Michael Trigoboff

I will deduct IQ points if you submit an assignment to me and your code prints out "assignment" or "your name." :-)

What To Upload

Each individual assignment will specify what you have to upload.

Do not get "creative" with the file names. Name your files *exactly* as specified (this includes whether letters in the name are upper-case or lower-case). I use scripts to compile and run your code, and when your file names are not what's expected, my scripts don't work. Annoying your instructor in this way will most likely not produce the benevolent and merciful attitude that you hope for...

Grading

Your grade for this course will be a function of your grades on the quizzes and programming assignments.

Quizzes

You are allowed to take each quiz *twice* (it will be exactly the same quiz each time you take it). Your grade for a quiz will be the *average* of your scores for the two attempts. This means that if you get a perfect score the first time you take a quiz, *there is no reason to take it a second time*.

Assignments

Some assignments will come in two flavors: **Regular** and **Premium**. **Premium** assignments are more difficult than **Regular** assignments. When the choice is available, you will need to do the **Premium** version if you want to get an **A** in this course.

To get a good grade for an assignment, you must do it in a satisfactory manner, which means your work must:

- be coded according to the [Coding Standards](#)
- compile without warnings (using gcc's `-Wall` flag)
- run without errors when given correct input and arguments. You do not need to check for input or argument errors unless:

- such checking is specified by an assignment
 - you feel like doing it for your own learning
- produce output *exactly as specified* by the assignment
(I run your code using scripts, and non-standard output causes problems in that environment)
- be submitted on time
 - for Assignments 1-3, there will be a 4-day “late submission” period after the deadline, during which the assignment can still be submitted. Submission during this period will incur a 10% lateness penalty.
 - there will be no late submission period for Assignment 4

Final Grade

Your final grade will be a function of the quality of your work on the assignments and quizzes.

To get an **A**, you will need to do **Premium** versions of the assignments when available. To get a **B**, you can do **Regular** versions of the assignments. To get either an **A** or a **B**, you will need to score 90% or above on the work you do.

This table shows how much each of the class activities contributes to your final grade:

Class Activity	Section	
	Classroom	Distance
Programming Assignments	45%	50%
Quizzes	45%	50%
Attendance	10%	—

This table shows the maximum possible scores for each assignment and quiz:

	Asgmt 1	Asgmt 2	Asgmt 3	Asgmt 4	Quiz 1	Quiz 2	Quiz 3
Regular	100	150	100	100	17	56	28
Premium	200		200	200			

You can use [this spreadsheet](#) to compute your final grade.

For more information, please go to the [PCC Grading Guidelines](#).

Classroom Behavior

I work hard to teach a deep and subtle understanding of how code works and how to write and debug it. This requires a classroom atmosphere of concentrated attention. The more of us who pay attention, the stronger and more focused that becomes.

Your behavior in the classroom has an effect on other students and on me. Your participation, in the form of questions and comments is welcome. Your attention is appreciated. But behavior that disrupts the atmosphere of the classroom will not be tolerated. This includes side conversations, texting, playing online games, etc.

I welcome your presence in the classroom. But if you choose to be in my classroom, you're required to be there mentally as well as physically.

Academic Honesty

Learning to write code is the mental equivalent of learning gymnastics. Cheating in a computer science course is like going to the gym and hiring someone else to lift the weights for you.

If you don't do the work yourself, you will not learn to write code. This will become painfully obvious the first time you have to write code that you can't find on the Internet.

It's fine if you want to work with other students on a project, but make sure that you do the work yourself and understand what you did.

Cheating and plagiarism will be dealt with as mercilessly as the college's [Academic Integrity Policy](#) allows.

Netiquette

There's a lot to say about "net etiquette," but short summary is sufficient: I expect us to treat each other and this course with consideration and respect.

I want this to be a safe learning environment. *There are no dumb questions.* What's dumb is to not ask when you don't understand something. By not asking, you're just wasting your tuition money and time.

The path to the top of the Mountain of Knowledge necessarily passes through the Valley of Stupidity. If you're not willing to feel dumb on the way, you're not going to learn much Computer Science. But in this class, we're not going to do anything to make it worse than it has to be. There's going to be plenty of opportunity to feel dumb, even if we're nice to each other.

Participation Expectations (Distance Learning)

Students in this course are expected to enter the course at least 3 times a week to work on the modules, check email, and participate in discussions. Prompt participation in discussions is especially important when group work is involved since others in your group are depending on your input. Each time you log into the course be sure to check for any new announcements, email and discussion messages, and calendar postings.

Content Required By The College

College Name

Portland Community College

Instructor

Michael Trigoboff
michael.trigoboff@pcc.edu
971-722-4393

Course

CS 201 – Computer Systems
4 credits
MR 11:00 – 11:50 TCB 212, 12:00 – 12:50 TCB 310
Prerequisite: CS 162 or equivalent. Recommended: CS 140U.

Attendance And Make-Up Policies

Students are responsible for all material covered in class.

PCC Policy Links

[PCC Registration Policy](#)
[PCC Payment Policy](#)
[Academic Integrity Policy](#)
[Student Rights and Responsibilities Handbook](#)
[Grading Guidelines](#)
[Course Drop/Withdraw Deadlines](#)

ADA Statement

PCC is committed to supporting all students. If you plan to use academic accommodations for this course, please contact your instructor as soon as possible to discuss your needs.

Accommodations are not retroactive; they begin when the instructor receives the "Approved Academic Accommodations" letter from you (submitted in person for courses on campus; via email for Distance Learning courses).

To request academic accommodations for a disability, please contact a disability services counselor on any PCC campus. Office locations, phone numbers, and additional information may be located on the [Disability Services](#) website.

Student Rights and Responsibilities Handbook

Students are required to comply with the [Student Rights and Responsibilities Handbook](#). The Handbook includes the Code of Student Conduct and the Academic Integrity Policy.

Title IX/Nondiscrimination

Portland Community College is committed to creating and fostering a learning and working environment based on open communication and mutual respect. If you believe you have encountered sexual harassment, sexual misconduct, sexual assault, or discrimination based on race, color, religion, age national origin, veteran status, sex, sexual orientation, gender identity, or disability, please contact the Office of Equity and Inclusion at 971-722-5840 or equity.inclusion@pcc.edu.

Non-Enforcement of Federal Immigration Law

PCC is a "sanctuary college." Find out more on the [resources for undocumented students page](#).

Flexibility

The instructor reserves the right to modify course content and/or substitute assignments and learning activities in response to institutional, weather or class situations.