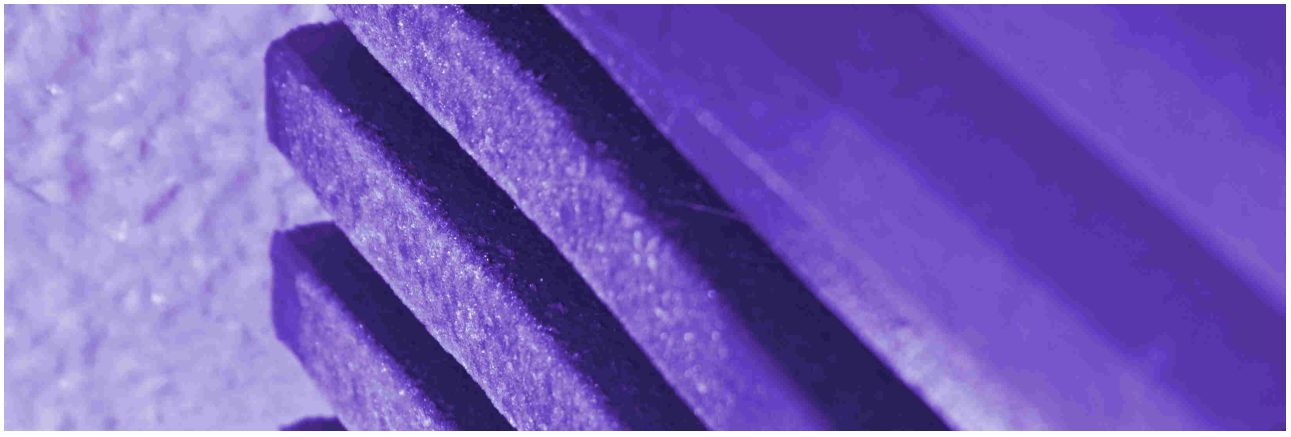


Introduction to Programming in Java - CIT 111

Course Syllabus: Spring 2019 | Instructor: Eric Darsow
Community College of Allegheny County

Course Info

Category	Content
Registration Classification	Number and Title: CIT 111 - Introduction to Programming Java Credits: 4 Hours: 4 hrs/week "lecture" Prerequisties: Basic Math
Catalog Description	This course is an introduction to program design, analysis and programming fundamentals using the Java language. Topics include the software development process, problem solving techniques, simple language basics, data representation and storage, program control structures, classes and their methods.
Prerequisites	Basic proficiency in essential math skills
Sections	< West Hills: Mondays/Wednesdays 10:00 - 12:05pm CIT-111-WH32 (0331887) Intro to Prog: Java
Instructor Information	Eric Darsow Computer Information Technology (CIT) Department Phone (shop): 412.894.3020 (Leave VM, please) Email: edarsow@ccac.edu
Course Materials	Technologyrediscovery.net is your home for this course. We will not use BlackBoard. The required textbook is the Liang9

Category	Content
Additional Reference Books!	**HeadFirst Java: O'Reilley publishes a "Brain-Friendly" technical guide series, and their Java edition has received many positive reviews from CCAC students.** This text, however, is very expensive to use as a course material (10c per page!), so it will not be referenced directly in this course. Amazon link I highly recommend Bruce Eckel's Thinking In Java which is available for $25 online. Bruce is committed to programmers collaborating and sharing, so he has released editions 1-3 for free online. I encourage you to have a paper copy of the book, but you can also download the entire 3rd edition from his website A more advanced reference-like text is The Java Programming Language, 4th Edition by Arnold, Ken, Gosling, James, Holmes, David. This book is endorsed by Sun, which is a developer of the Java language itself. For beginner programmers, many concepts here will be explained with more depth than needed, but it is good to be exposed to the deeper technical explanations of the language.

Learning Outcomes

The following content is extracted directly from the CCAC master course syllabus for CIT 100:

This course is an introduction to program design, analysis and programming fundamentals using the Java language. Topics include the software development process, problem solving techniques, simple language basics, data representation and storage, program control structures, classes and their methods.

1. Use a Java integrated development environment (IDE) to design and implement applications on a computer.
2. Apply basic problem-solving techniques to novel situations that are relatively similar to models introduced in the course.
3. Use various data types by implementing them in the correct operations.
4. Create simple graphical user interfaces (GUI) using Swing JOptionPane components.
5. Use control structures in a Java application.
6. Utilize class methods found in the Java application programming interface (API).

Course Philosophy

Creativity is a core goal!

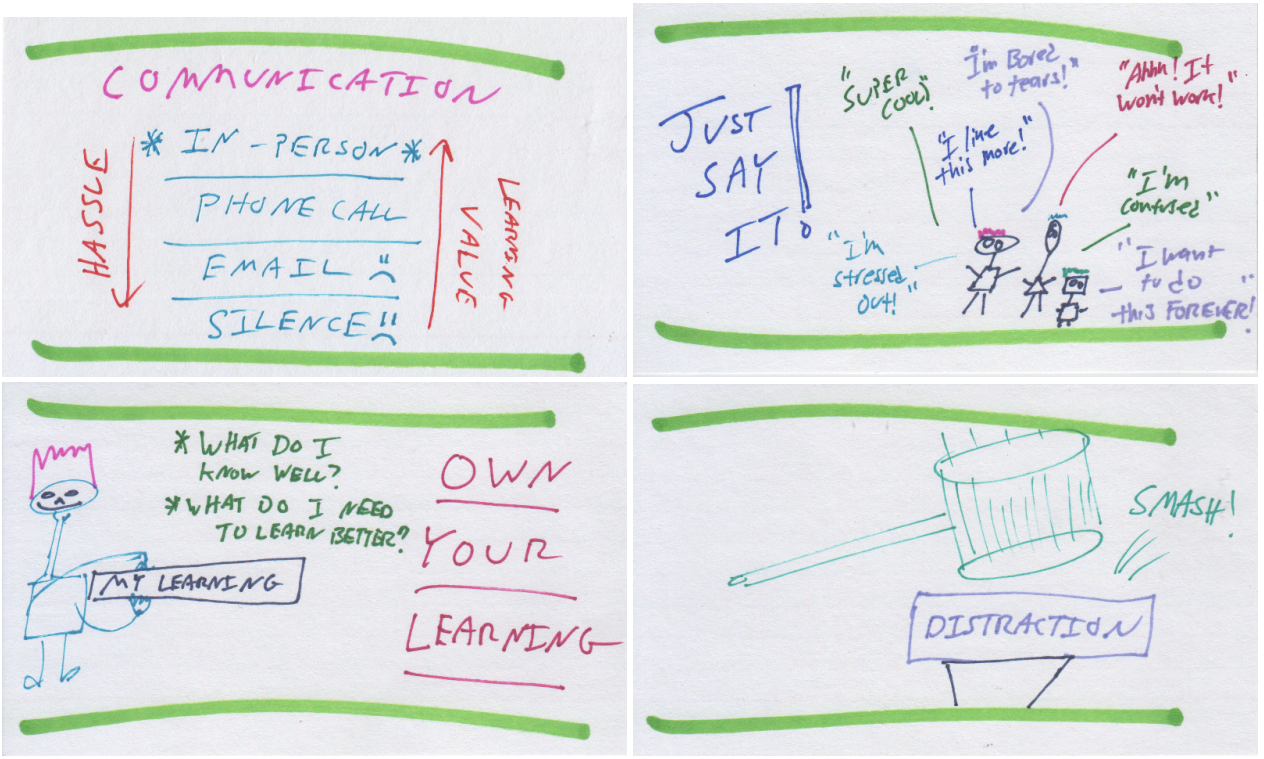
This is an introductory programming course designed for students with little or no programming or computer science background. The mission of the course is to cultivate a **genuine interest in what makes computers tick** and how to communicate with the machines in a useful and creative way. Curiosity and interest in the core concepts of Java and object oriented design will be developed during the course as students are supported in approaching the learning process with openness and eagerness.

We can LEARN to be resilient learners

For a vast majority of students in an introductory programming course, the learning process will involve both organic and exciting discoveries as well as struggles, setbacks, and frustrations. Resilience, therefore, is essential in this process. Thankfully, **resilience can be developed with practice.** The course activities are designed to support the development of resilience—or bouncing back and pushing onward—in the endeavor of computer programming.

Course Norms

We can consider our class like a mini-community of folks who are working on a project together, in this case, we're learning about computer fundamentals. I've done some thinking about how we can work together and stay happy while doing so, and propose the following group norms which hopefully can become shared values by class members. To avoid blobs of text, I've made little cards describing the norms so we can discuss them in class.



Practice / "Homework" structure

As an introductory and overview course, our most important time together will be **during class sessions.**

"Late work"

The order in which we learn stuff certainly matters a lot, but exactly when a particular chunk of learning happens doesn't impact the quality of that learning. With this principle in mind, I accept all assigned work up until the day before CCAC's college-wide grade submission deadlines (for mid-term and final).

Keeping pace with the class, however, is likely to be very important to your learning process. For example, what we learn in chunk 4 is an extension of what we we'll study in chunk 3. If you have put off doing out-of-class practice, you won't be comfortable with chunk 3 stuff and you'll likely be confused by the new content, and this is do doubt frustrating for everybody involved. So do yourself a favor, and, when possible, schedule out regular time to practice our class skills and work through exercises.

For in-person sections: Missing class sessions

Come to class! We want you to be part of the group! I'm here to help you learn stuff, so communicating with me about your class attendance is important so I can help keep you on track with the course.

If you miss class, **please find some time in your life to work through whatever exercises have been posted for the day you were absent.** Make note on what you would like more support to learn. Then come to office hours (posted on our course website) and ask questions or work through stuff that you couldn't figure out.

If you miss class and I don't hear from you, I'll likely drop you call and check in to see how you're doing. If you've missed 2 class sessions and I haven't heard from you via email or phone, I'll probably call you to check in, so best to save my contact numbers in your phone. 412.894.3020

Core Habits for Meaningful Learning:

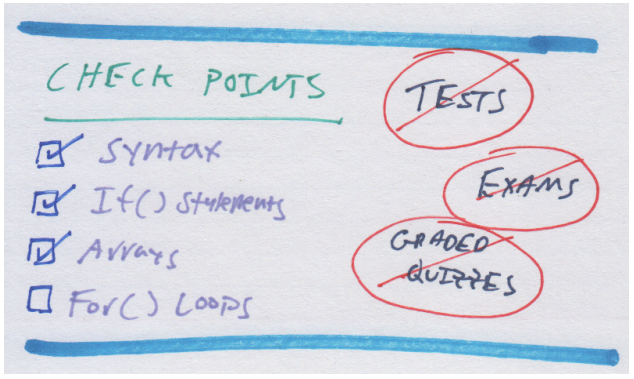
1. **Deliberately practice developing your skills independently** and/or in groups – outside of class – a few times a week. Practice distraction free: silence phones, use music sparingly, and work somewhere you can focus.
2. Think of **computer tools as instruments** that can create their version of "music." Treat them as such and learn your instrument's ins and outs. Learn how it works on the inside as much as you can—even if it

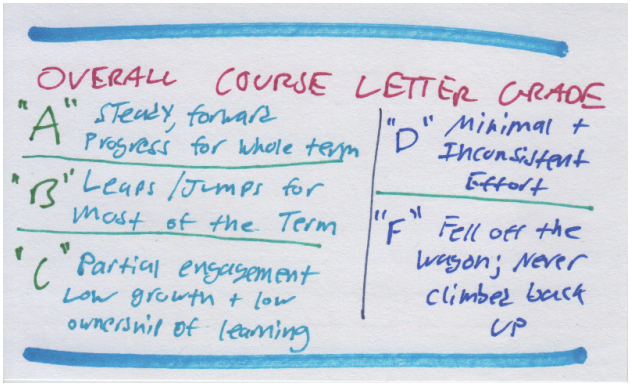
doesn't seem immediately relevant. It will pay off big time in the future when you have to get it to do new things.

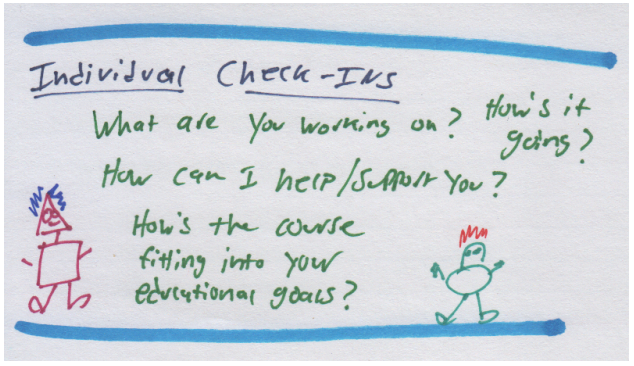
3. **Monitor for yourself what you don't yet understand** or can't yet do and systematically exert effort to understand that stuff or do those things.
4. **Tinker with your tools, and break stuff** (code, applications, operating systems, networks) and figure out how to fix it. This is the advantage to learning computers—breaking them doesn't cost anything like it does in, say, woodworking, where messing up means wasting wood.
5. **Honor important waypoints** in your learning process. ("When it really counts, make it happen.") by being fully prepared for checkpoints and work sharing events presentations, etc.
6. **Make explicit note of any learning aids you use ("cite your sources")** during any work you do in this course. Include links to code you heavily reference and make happy note of major breakthroughs you made on your own.

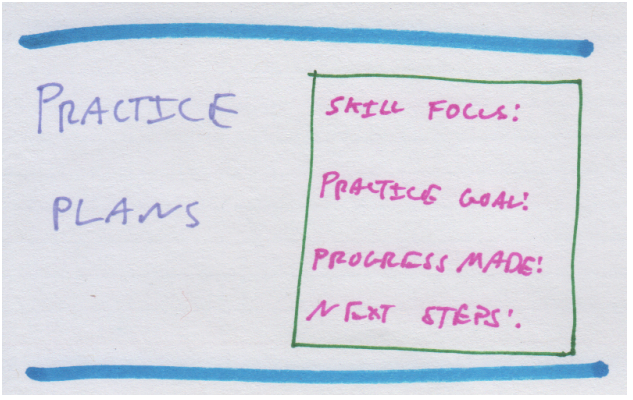
Assessment of Learning Progress

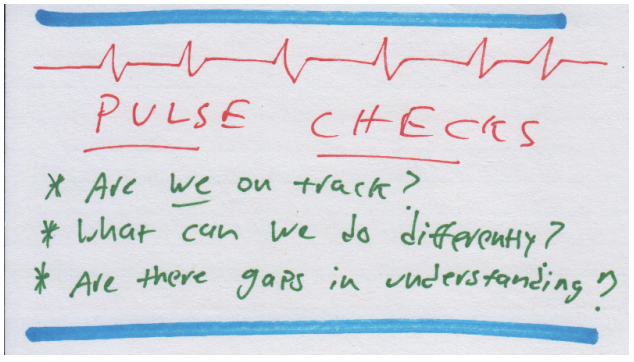
High-stakes tests, exams, quizzes that make and break grades have generally given the notion of assessment in education a bad flavor in many folks' mouths. We shall, in this course, attempt to take back assessment as a useful tool for figuring out what we know, and what we have yet to learn, and aligning the course to make that happen. Here are some cards describing some assessment mechanisms we'll use in this course:

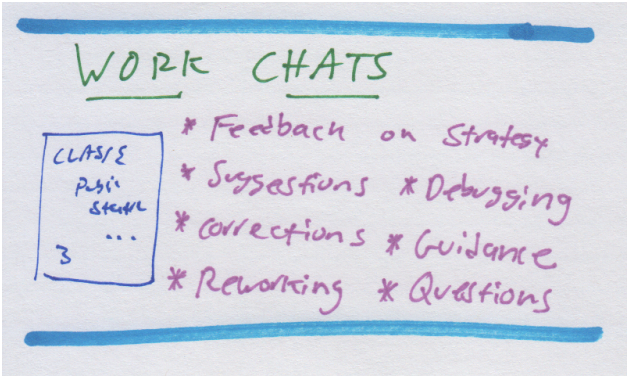












Checkpoints (not tests!):

No heavily weighted tests

This course will not involve tests that are high pressure, completed entirely independently, and heavily weighted. Rather, the course is designed to encourage and support regular practice which leads to incremental learning. **Many courses that heavily weight tests end up encouraging students to cram in content learning directly before each major exam,** and this often detracts from the regular practice that is likely much more important to becoming a proficient computer user.

What is a checkpoint?

In this spirit, **we will have a few checkpoint tasks that are completed during class and involve completing a mini programming project that you have not previously seen.** The goal of the checkpoints is to practice your evolving skills and to communicate with me about your learning progress so I can support you as much as possible in meeting your educational goals. They should be prepared for using the preparation guides on the course website, but they are not designed to be stressful or "make or break" in quality. They are literally a checking in about your progress through learning Java. Nothing more, nothing less.

The mini-project for the checkpoints **should take approximately 2 hours to complete** (the entire class session) and will demand that you apply all of the core concepts learned in the course up to that checkpoint time. (Languages are by definition cumulative, so what we learn during the second class period, for example, will be required for all future classes and checkpoints).

What resources can I use during checkpoints?

During the checkpoint, **you will be encouraged to use any and all programming resources that you find helpful,** including the Java API documentation and community forums like StackExchange. Of course, your use of these resources must be cited appropriately in your code—through comments—just as you should always do, forever and ever, throughout your programming journey.

I will be around to provide guidance during the checkpoints (not for online sections, sorry), but will be mindful of creating an experience that checks your learning so far, not mine. So I may not answer questions with the same degree of explanation as I might during a normal class session.

You will receive written or verbal feedback on your checkpoint program within a week of its completion.

Course Letter Grades

Core assessment principle: The most important outcomes of the course is the degree to which students are **comfortable implementing their new skills and knowledge in "real world" situations** (i.e. to solve meaningful problems). This will be assessed through a review of the work you post on our course upload repositories.

This course uses a standardized grading procedure that aligns with CCAC's college-wide grading policies which require teachers to issue each student a cumulative course grade of A,B,C,D or F. Both the student and faculty handbooks associate each letter grade with the following performance-related words:

- A – Superior (4)
- B – Above Average (3)
- C – Average (2)
- D – Below Average (1)
- F – Failure (0)

Suggestions for successful technical learning

We are working to learn how to use a computers—really fancy adding machines—to help us do stuff that's useful like organize, store, and present data. Since computers are very complicated, successful learning should adapt to this environment. The following principles can be thought of as suggestions for building your computer knowledge and are based on my experience learning many new technical topics from scratch over the past few years:

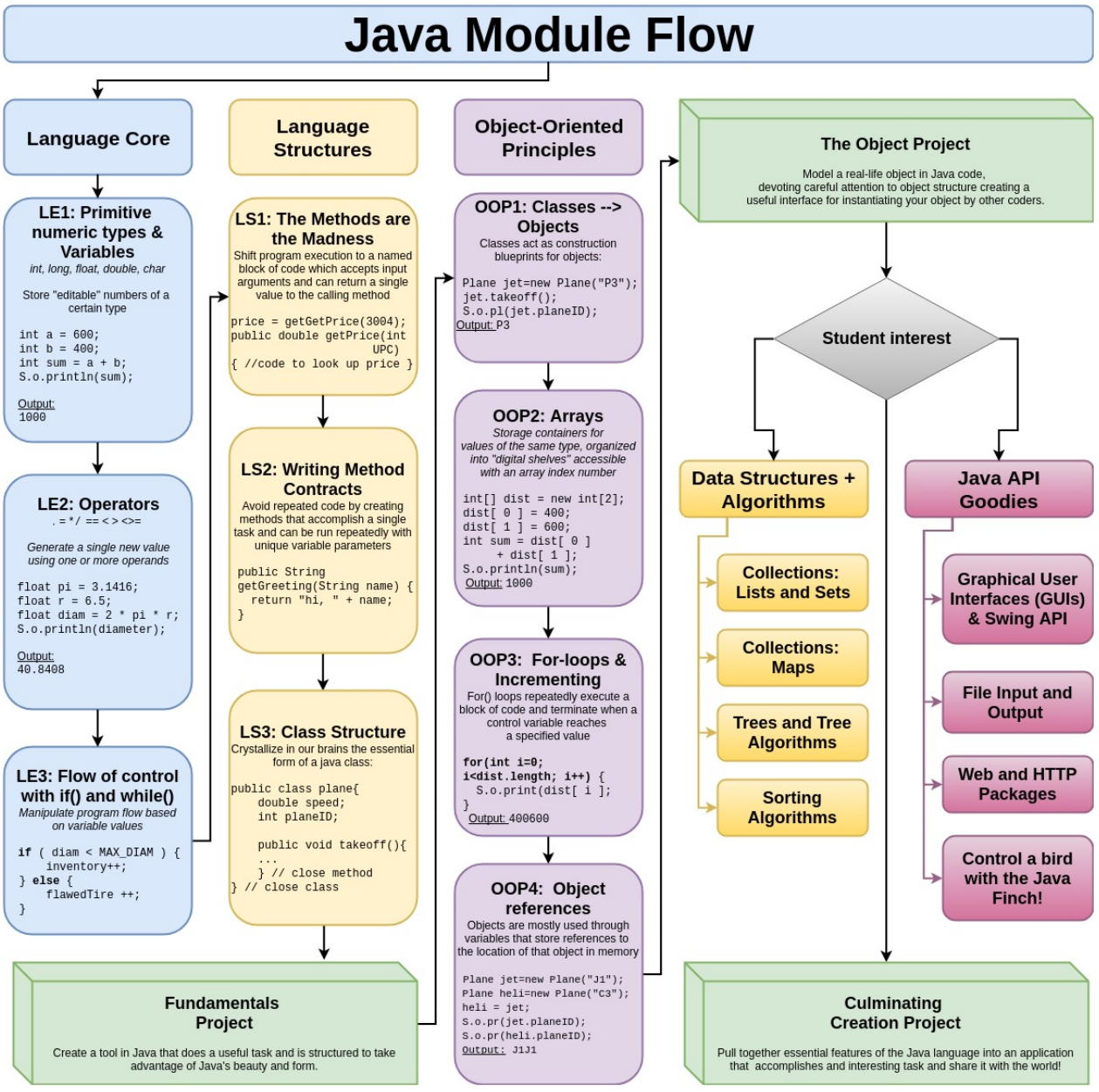
- **Find a book, tutorial set, or other learning resource** that applies to the topic you're studying and devour it all. Avoid skipping around and switching resources once you have found one that works with

your learning style. This will help maintain a steady learning path and increase the quality of your knowledge.

- **Locate the technical documentation** for any tools you are using and study it carefully. These materials are usually written and published by the creators of the tools themselves. Become intimately familiar with the way the documentation is structured so you can find the information you need quickly when the moment arises.
- **Expect your learning to spiral upward** as you learn more and more about the important topics that are connected to your study. Since topics are often interrelated and complex, expect that you'll need to be exposed to the same topic or idea a few times before it soaks in.
- **Find a mentor or friend** whose a little more proficient in the system than you are—as him/her/them questions often. Soak up what they know whenever possible. Work together when possible.
- Challenge yourself **by not asking for help the moment you get stuck—tinker a bit.** Experiment with a solution and see if it works. Working through confusion is an incredibly meaningful step in technical learning, since once you're doing paid work, you're expected to know how to solve problems.

Course learning cycle model

The course is organized into four components which are high-level learning categories composed of sub-components called modules. Each module evolves through four distinct phases: introduction, guided practice, independent practice, and feedback on final products. Once each module in a component has been mastered, students are ready to plan and implement a culminating project. This process is visually represented in the following flow chart:



back to top

Accommodations for Individuals with Disabilities:

The college recognizes its responsibility to provide academic and nonacademic services and programs equally to individuals with and without disabilities. To this end, the college provides reasonable accommodations for qualified students and employees with documented disabilities consistent with the requirements of the Americans with Disabilities Act, sections 503 and 504 of the Rehabilitation Act and other federal, state and local laws and regulations. The college maintains an Office of Supportive Services at each campus location to receive, review and evaluate requests from students who require an accommodation with respect to their educational program. Students' requesting reasonable accommodations due to a documented disability must first register with their campus' Supportive Services Office and obtain an official letter identifying approved accommodations to be distributed to their faculty members.

Attendance Procedure for Pregnancy and Pregnancy Related Conditions:

In accordance with Title IX of the Education Amendments of 1972, absences due to pregnancy or related conditions, including recovery from childbirth, shall be excused for as long as the absences are determined to be medically necessary. Students will be provided with the opportunity to make up any work missed as a result of such absences, if possible. For more information or requests for accommodations, students should inform their instructor(s) and/or contact the Civil Rights Compliance Officer/Title IX Coordinator, Sumana Misra-Zets, at 412.237.4535 or smisra@ccac.edu.

Attendance Procedure for Religious Observance

The college will make reasonable efforts to accommodate students who must be absent from classes or miss scheduled exams in order to observe a religious holiday or participate in some other form of religious observance. Students shall be provided, whenever possible, reasonable opportunity to make up academic assignments missed due to such absences, unless doing so would create or impose an undue burden on other students or the College. It shall be the students' responsibility to provide written notice via the Request for Accommodation for Religious Observances Form (accessible at https://www.ccac.edu/nondiscrimination/) to every instructor for each course in which an accommodation is being requested. For more information contact the Civil Rights Compliance Officer/Title IX Coordinator, Sumana Misra-Zets, at 412.237.4535 or smisra@ccac.edu.

Chosen First Name Procedure for Students

Many individuals use names other than their legal first name to identify themselves for a variety of personal and/or cultural reasons. The college seeks to provide an inclusive and non-discriminatory environment by making it possible for students to use a chosen first name on college records when a legal name is not required. Chosen first names may not be applicable in certain programs due to the requirements of accreditation organizations and clinical sites. For more information, please see the Student Handbook

back to top `