CISC 190: Java Programming

CRN 23337 Online Spring 2020 | Instructor: Tasha Frankie tfrankie@sdccd.edu (use Canvas Inbox please)
Office Visiting Hours: M 9:00-9:30am BT-211U | W 8:00-9:30am BT-211U | MW 12:45-1:15pm BT-210
Tu 9-10am and Th 8-9pm "virtual" chats via Zoom

Get Started:
Install Netbeans and Dr. Java; start using bookClasses to draw with a Turtle. Find a study buddy to encourage you through this course!

1. Read and Try:

Type up code from the reading, and try it out. Increase comprehension and have more fun by working with a study buddy.

2. Discuss:

Analyze sample code in specially designed questions on Perusall. Enhance understanding by identifying and discussing main concepts from the reading.

3. Code-Writing Practice:

Utilize new module concepts along with previously learned concepts to solve a programming problem. Maximize learning by discussing approaches with a study buddy and comparing code after you've written your own solution.

4. Technical Interview Practice:

See how well you understand the new concepts - can you write a solution quickly, like you need to do in a technical interview?

5. Explain:

Write code that utilizes module concepts and record a short video explaining the purpose and mechanics of the code. Demonstrate expertise by connecting concepts to other material - you can be a YouTube programming guru!

6. Reflect:

Reflect on your experience with this module. Boost independent learning skills by identifying successes and challenges.

```
while (there are more modules)
if (need more practice)
else
Done with modules
```

Final Project:

Apply the concepts you've learned this term to a project you design. Showcase your programming mastery.

List of Modules

1. Turtles: Using Methods and Parameters
2. Turtles: Writing Methods
3. Pictures: Loops and Arrays
4. Pictures: Nested Loops
5. Pictures: Conditional Execution
6. Classes: Creating Your Own Custom Class
7. Advanced Topics
   - Classes: Inheritance and Polymorphism
   - File I/O
   - Exception Handling
   - GUIs

Common Student Questions

Module	Points Available
0 - Getting Started	25
1 - Turtles: Using Methods	72
2 - Turtles: Writing Methods	102
3 - Pictures: Loops and Array	129
4 - Pictures: Nested Loops	121
5 - Pictures: Conditional Execution	149
6 - Classes: Creating Your Own Custom Class	113
7 - Advanced Topics	85
8 - Final Project	250
Extra Credit	30 points max

Note: Point values deliverables are subject to change. See Canvas for due dates and point values.

How do you assign letter grades for the class?

This course is graded out of 1000 points, with slightly more than 1000 points available to earn. Your grade is based on successful completion of learning opportunities designed to help you master the objectives of this course. A final grade of C or better indicates that you have the ability to successfully apply the theory and techniques taught in this course to subsequent courses and in practice.

Grading Scale:

A = 900+, B = 800-899, C = 700-799, D = 600-699, F = < 600

How do I earn extra credit in this class?

Extra credit is earned several ways:

- Attend special events (dates and times announced on Canvas). To receive credit for attending an event, sign in at the event and write a short summary that you submit on Canvas in the Extra Credit module.
- Learn more about computer science and its applications. Propose a project and get it approved (e-mail the proposal using the Canvas Inbox). Extra credit projects do *not* need to involve writing code; reading and summarizing an article related to computer science could be sufficient.
- Complete extra code-writing practices (does not count against the 30 point maximum).
- Attend Peer Mentoring sessions throughout the term.
- Propose alternate methods for earning extra – they just need to be approved!
- Watch Canvas for additional extra credit opportunities.

What textbooks/materials do I need for this class?

You do not need to purchase any textbooks for this class. We will utilize a combination of Open Educational Resources (OER) and online tutorials.

Additionally, you will need:

- Access to a computer that can run Dr. Java and NetBeans. Computers in the Mesa library, including the STEM Center, have these programs installed.
- A way to make videos with audio. Suggestions: attach a microphone to your computer (or use a built-in microphone) and do a screen capture, or use your smartphone to record a video.

I've never programmed before – should I take this course?

Yes, you should! I assume that students have never programmed before – we will start at the beginning. If you have a basic ability to use a computer (e.g. type, follow instructions to install programs, manage storage of your files, etc.) you can be successful in this course. Just make sure to keep up with the course and be proactive about asking for help (ask sooner, rather than later, so you don't fall behind!).

How does an online class work?

I've designed this online class to provide you with many learning opportunities, including discussing the material with your classmates and practicing applying what you've learned to fun problems. The graphic on the previous page describes the learning activities you will find in each module. You should expect to spend 10-15 hours each week working on the course; I recommend dedicating a few hours each day.

You may work ahead at your own pace for most of the course, but you need to keep up with each module's due date. Late work is graded at the discretion of the instructor, and may be subject to late penalties and/or be given a numerical grade without feedback. You are encouraged to make corrections to all graded work to earn more points; for maximum learning, corrections for each module should be submitted by the due date of the next module.

There are several important deadlines for this course:

- *Tuesday, Feb. 18 (Add/drop deadline). Complete all of the "Getting Started" and "Using Methods and Parameters" modules.* If you do not complete these modules by Feb. 11, I will drop you from the course. In my experience with students in previous terms, you are not likely to successfully complete the course if you haven't successfully gotten started by the add/drop deadline.
- *Friday, May 29. Complete the "Final Project" module and any resubmissions.* This gives me time to grade and to contact you before grades are due if there are any issues regarding your final project submission.

More Common Student Questions

Help! I'm stuck! What should I do?

Great question! I never want you to feel "stuck" for more than ***15 minutes*** before asking for help. Here are some suggestions for places to find help:

- Ask in the Q&A discussion forum on Canvas—I encourage everyone to post questions and answer questions frequently.
- See if some of the other learning opportunities in the module are helpful in clearing up your confusion.
- Seek out free tutoring in the Mesa library.
- Form a study group with your classmates.
- Attend Peer Mentoring sessions.
- You are always welcome to come to office hours (in person or virtual) or to e-mail me using the Canvas Inbox at any time. Please use the Canvas Inbox to ensure that I see your e-mail; e-mails sent to my regular e-mail inbox are highly likely to get lost in the flood of e-mails I receive on a daily basis.

What is Peer Mentoring?

Peer mentoring is a way for you to get extra practice for this course. Several times each week, a peer mentor will lead a session designed to help you with specific skills related to this course. Everyone can benefit from peer mentoring; We will attempt to make peer mentoring sessions available via Zoom. See Canvas announcements for more details.

What is the collaboration policy?

To maximize your learning, I recommend that you form a study group and work with your classmates. Please list the names of anyone you receive help from in the comments of the code you submit (regardless of whether the person assisting you is enrolled in this course). When working in groups to create code, you may want to try pair programming, a technique proven to increase student understanding in introductory programming courses; just be sure to switch roles frequently. If you use pair programming to complete an assignment, it is acceptable to submit identical code. The exception to this collaborative policy is in the "Explain" portion of each module; you need to write and present the code individually. You are welcome to get feedback from your peers prior to submitting the assignment though!

Can I improve my grade on an assignment?

Absolutely! It takes a lot of time to provide feedback to you; I'd like you to read my comments and make corrections so you learn from your mistakes. As incentive for you to rework an assignment, you can earn more points for any corrections you make and resubmit. For maximum learning, corrections for each module should be submitted by the due date of the next module.

What skills am I learning in this course?

In addition to learning Java programming, you will also have the opportunity to practice and refine soft skills that are in demand by employers. Did you know that employers are often less concerned about what programming languages you know (these skills are easy to learn on the job), and are more concerned about the soft skill set that you bring as an employee? Here is a list of some of the soft skills that you will refine in this course; please feel free to put them on your resume! Skills: critical thinking, problem solving, analytical skills, troubleshooting, research, planning, creativity, written communication, teamwork/collaboration, resilience/perseverance, lifelong learning.

Required Administrative Stuff

Course Description: This course is an introduction to programming using Java. The course covers the fundamentals of object-oriented programming utilizing the Java programming language for general purpose business programs and interactive World Wide Web-based Internet programs.

Course Learning Outcomes:
Students will be able to:

Understand from a Java perspective the control of program flow based on the conditional evaluation of a boolean expression.
Use Java to provide an if… else structure as a solution to a software problem.
Understand from a Java perspective the control of program flow with a looping structure.
Use Java to provide a looping structure as a solution to a software problem.

Student Learning Objectives:
Upon successful completion of the course the student will be able to:

1. Design, compile, and test Java programs that display objects, patterns, and words.
2. Create methods that carry out tasks using Java programming.
3. Employ the use of blocks within a method to create Java programming codes.
4. Diagram flowcharts for the purpose of constructing decision structures, accepting keyboard input, and nesting "if" statements for Java programs.
5. Create and test Java programs that use arrays.
6. Apply knowledge of general classifications to more specific objects using the inheritance principle.
7. Employ the abstract method to create arrays of subclass objects in Java programs.
8 Apply inheritance concepts to create Graphical User Interface widgets.
9. Use an event-driven Graphical User Interface framework to create interactive programs using layout managers.
10. Employ exception handling as an object-oriented technique to manage program errors.
11. Design Java programs that employ the use of file classes to input and output data for the programs.
12. Describe multithreading and list the components of the thread lifecycle.

Canvas:
We will be using Canvas this term. Course handouts, including, but not limited to, this syllabus and programming assignments, are available via Canvas, located online at https://sdccd.instructure.com If you are not familiar with Canvas, some orientation material can be found in our course after logging in, and additional training is online at http://sdccdonline.net/students/training/. Additionally, there is a 24/7 Helpdesk for technical support via telephone at 844-612-7421 or e-mail at support@instructure.com. You are also welcome to ask the instructor for help and/or to direct you to orientation material.

Student Code of Conduct:
Students are expected to adhere to the Student Code of Conduct at all times. Students who violate the Student Code of Conduct may be removed from class by the faculty for the class meeting in which the behavior occurred, and the next class meeting. This included, at a minimum, behaving respectfully toward other students and the instructor, both in the classroom and online.

Students will receive a 0 for any work missed during a removal period. Additional work, including written or verbal apologies or reflective essays may be required before returning to class.

Incidents involving removal of a student from class will be reported to the college disciplinary officer for follow up.

The Student Code of Conduct can be found in Board of Trustees Policy, BP 3100, Student Rights, Responsibilities and Administrative Due Process posted on the District website at: http://www.sdccd.edu/public/district/policies/index.shtml.

Cheating/Plagiarism:
Students are expected to be honest and ethical at all times in the pursuit of academic goals. Students who are found to be in violation of Administrative Procedure 3100.3 Honest Academic Conduct, will receive a grade of zero on the assignment, quiz, or exam in question and may be referred for disciplinary action in accordance with Administrative Procedure 3100.2, Student Disciplinary Procedures.

Accommodating Students with Disabilities:
I have made every effort to make this course accessible to all students, including students with disabilities. If you encounter a problem accessing anything in this course, please contact me immediately.

Students with disabilities who may need academic accommodations should discuss options with their professors during the first two weeks of class. All information will be kept confidential. A written notice of registration with DSPS is required for accommodations to be made.

Students that need evacuation assistance during campus emergencies should also meet with the instructor as soon as possible to assure the health and safety of all students.

For more information, you may contact the DSPS Office at Mesa or the website at http://dsps.sdccd.edu.

Maintaining Enrollment in the Class:
It is the student's responsibility to drop all classes which he/she is no longer participating.

It is the instructor's discretion to withdraw a student due to excessive absences or failure to complete the modules by the due dates.

Students who remain enrolled in a class beyond the published withdrawal deadline, as stated in the class schedule, will receive an evaluative letter grade in this class.

Failure to submit work in a timely fashion prior to the add/drop deadline will result in the instructor dropping the student. Space in the class will then be given to a student on the waitlist. After the add/drop deadline, the instructor is not responsible for dropping any student – it is the student's responsibility to drop if he/she is no longer attending. However, it is at the instructor's discretion to drop students who are no longer participating in the class at any time.

The instructor reserves the right to change the contents of this syllabus as needed during the term. If the syllabus is changed, an updated copy will be posted on Canvas.