Cpt S 321, Spring 2023

Instructor: Dr. Venera Arnaoudova, Associate Professor at EECS.
Email: venera.arnaoudova@wsu.edu
Webpage: www.veneraarnaoudova.com

Class Website: On Canvas

Class time: MWF 12:10-1:00pm.

Pullman: ETRL 101
Everett: EVRT 458 (Streamed from Pullman)

You are expected to attend all lectures in person unless there is a health emergency.

My Office hours (in person and virtual): MWF 1:00-2:00pm (i.e., right after class).

Virtual: There is a waiting list so please be patient – I let everyone on a FIFO basis**.**

Physical location: EME 127

Virtual Link: https://wsu.zoom.us/j/93419173425?pwd=bHZYcDhsYjJvSzMwYUloRVFEUlV3dz09

Meeting ID: 934 1917 3425

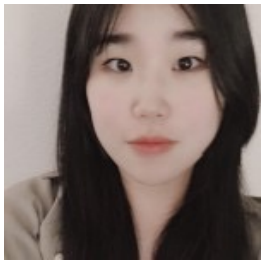
Passcode: 855280

Who am I?

- Associate Professor at EECS
- Joined WSU in 2015
- Ph.D. from Ecole Polytechnique de Montreal (Canada)
- B.S. from Polytech'Lille (France)
- Teaching at WSU: CptS 321, CptS 581, CptS 582; In the past: CptS 422
- Research at WSU:
  - Head of the Software Engineering Lab (SEL). Interested in gaining UG research experience? Come and talk to me!
  - Our research is about program comprehension and improving developers' productivity
- Other: English is my third language

My current Ph.D. students

Devjeet Roy



Ziyi Zhang

How to reach me and the TAs

- 3 UG TAs (Eli, Manjesh, and Taylor) this semester
- Contact information for me and TAs will be available on the Canvas course webpage -> Syllabus
- Make sure you check that before submitting an assignment or before sending an email
- **Important**: When sending emails, please cc me and all TAs every time!

Communication

- Please post all course-related questions on Canvas so that everyone has the same information and we are all on the same page. There is a class **"Discussions"** board with different forums; makes sure you subscribe to all of them.

- Me and the TAs will be posting also in the "**Announcements**". Make sure you subscribe to it!

- Emails: please include me and **ALL TAs in ALL your emails**
  - Have "CptS321" in the email subject.
  - Be respectful and professional (At a minimum, **start with "Hi"** and **finish with "Thank you" or "Regards" and sign with your name**).
  - For grade-related questions, please email **ALL TAs** and cc-me.
  - If you need to send me an email regarding a personal matter, send it to me only but feel free to cc the TAs if you feel comfortable doing that.

The Challenges of This Course

- Primarily learning to develop a larger scale software project
  - Assignments build incrementally
- Learning C#
  - For those that don't know it, it's actually a pretty nice language
- Learning/practicing a variety of topics to flesh out your skillset in preparation of the 400 level courses and internships
  - High quality design, testing, version control, GUI, threading

Books (for C#)

- Optional book for the course ("C# in a Nutshell") should be on the shelf at the WSU bookstore (Check Amazon if you prefer the Kindle version)
  - If you're not good at self-guided learning and using internet sources by this point then you'll definitely want to buy the book help you learn C# faster
  - Note that even if you DO buy the book, you'll be expected to do reading from various online sources throughout the semester. In other words the book doesn't have every last bit of information that you'll need.

- Essential C# Book: https://essentialcsharp.com

- A lot of other topics that are not in the books

Syllabus

- Syllabus is posted on Canvas

- It is the “constitution” for this course and you are responsible for reading and understanding everything within it. Consider reading it to be a **homework assignment to be done by next class at the latest**. If you have any questions/doubts we will discuss them next time.

- This introductory lecture only gives an overview of the course but does not cover every last policy, so you must read the syllabus.

Topics

- C#:
  - Everything we do will be in C#. You cannot write C++ code in this course and expect to get credit for it.

- More data structures and algorithms (yes, there's plenty more than what was covered in 223).

- Flexible, efficient, maintainable design. Another way to state this is that we'll be focusing on "good design". This is a technical skill, but is somewhat less of an exact science.
  - Organizing your projects
  - Naming
  - Quality of test cases
  - The use of appropriate design patterns

Attendance

- Unless you are sick or it endangers your safety to come to class, then you are expected to attend.

- **Being present in class accounts for 10 points of your final grade** (out of 200). **An absence will cost you 1 point** each time **unless** the instructor is made aware of your inability to attend class. It is the responsibility of the student to provide a written explanation for the absence to the instructor as soon as it is reasonable to do so. When possible, students should provide appropriate documentation for their absence. Students who attempt to gain advantage through abuse of this policy (e.g., by providing an instructor with false information) will be referred to the Center for Community Standards for disciplinary action.
  - **There is a Google form survey for every class that you ALL must fill in by the end of each class.**

- Missing a class (for any reason) implies that you'll need to watch the recording to catch up.

Why Attendance is Important

- Slides will be posted online.
- Solutions to in-class exercises will **NOT** be posted online.
- **Submitting the coding exercises that we do in class brings you bonus points!**
- Focus more on listening to the lecture than trying to transcribe everything I say, draw or type in the lecture. If you just copy things down without understanding them it's not likely to help you much.

Course Expectations

- **Prerequisites:** CPT S 223 with a C or better
- There is a **LARGE** amount of independent learning expected in this course
- Read all notes posted online AND any pages they link to AND any other documentation you need to complete your task

What You “Should” Know Now

- (list is not exhaustive)
- C and C++ language syntax (or Java would also suffice)
  - Reminder: we’re coding in C# in this class, but we’ll relate certain language features to C++/Java to help with the transition
- Data structures: stacks, queues, trees, hash tables, graphs
- Time complexity analysis
- Object-oriented programming and concepts such as inheritance and polymorphism

How to Succeed

- Whenever you run into something you don't understand, **ask questions**. It is almost always the case that the students asking questions constantly end up doing well and the ones silently falling behind without seeking help when they need it end up doing poorly.

- **You can earn up to 20 bonus points (equivalent to 2 HW assignments) for participating in class!**
  1. **working on the in-class exercises and submitting your code to the EECS GitLab repo** (you must create a separate repo for in-class exercises – do not use the repo for the HW assignments; for your coding exercise to be considered for bonus points, you must tag it by the end of class unless otherwise stated) and
  2. **being interactive, i.e., ask and answer questions in class or on Canvas**.

- More on interacting: Ask questions (voice/Canvas), try to answer questions (I give points for trying even if your answer is wrong), show your code during the in-class exercises, comment on shown solutions.

- Get help from instructor, TA, tutor, or whatever other source you need to make sure you never fall behind.

- Do all homework assignments. There is ~ one per week and they are designed to help keep you on track.

- **If you do all the homework assignments and class exercises and you understand the concepts we talk about in class, you'll be in good shape for the exams.**

Homework Format

- Weekly programming assignments will be posted on Canvas. The requirements for the assignment will be listed. Read them carefully or you may end up missing a requirement and losing points. **Check the grading schema at the end of the file.**
- The homework assignments aren't intended to be busywork, but given the amount of material we cover and what it takes to demonstrate your understanding of it, they might end up being quite time consuming.
- **IDEs for our desktop application**
  - **Windows desktop application (i.e., WinForms development): Visual Studio (VS) Community Edition** is available for free. Supported on Windows only, so you'll need that too
  - **Cross platform desktop application: JetBrains Rider** – login with your wsu email.
- **HW Submission through Canvas and EECS GitLab**: https://gitlab.eecs.wsu.edu. **Me and the TAs should be added as "maintainers" to your repository.** Different assignments will be submitted in different branches. An assignment is considered submitted if and only if it is **TAGGED on the EECS GitLab repo AND the link is submitted on Canvas**! Check the syllabus and the turn in guide for more details.

Homework Format (cont.)

- You must follow the directions in the assignment. You can't just make up your own and claim you've demonstrated the relevant level of knowledge/skill.

- Most likely way to lose points is probably to not really understand the material. But the second most likely way is to misread or ignore the directions.

- **You can discuss concepts and algorithms with other students** (and you are encouraged to do so!), BUT **do all the coding yourself**. If very similar assignments are submitted, then all students involved will get an F for that assignment and for the course, and will be reported to the office of student conduct.
  - I have given students an 'F' for cheating before and reported them to the office of student conduct. I'd rather not do it again. Do your own work.

Grading

- See the syllabus for a breakdown of the grading scale. The short story is that there are 200 points divided as follows:
  - 110 points (~50% of your grade) comes from homeworks (each accounts for 10 points)
  - 80 points comes from exams (mid-term 1, mid-term 2, and final)
  - 10 points for being present in class (FILL IN THOSE GOOGLE SURVEYS AT THE END OF EACH CLASS!!!)
  - **BONUS points**:
    - 20 for participation in class
      - Asking and answering questions, commenting, participating in Canvas forums, etc. Make sure you put in the Google survey how you participated in class each time.
      - Submitting the in class exercises (Submitted = TAGGED + submitted link on Canvas, more in the turn in tutorial)
    - 10 points for an in-class presentation (see syllabus for details)

- Syllabus on Canvas has a calendar with ALL homework due dates and exam dates. I will **not** send reminders – I trust that you are independent.

- Check your progress every few weeks and if you're below a B-, you need to change something or else you might not pass the class. **Talk to me and the TAs!**

- **The TAs will provide prompt feedback on your HW assignments – make sure you understand and incorporate that feedback in the remaining assignments. Grades become solidified if no concerns are raised within 4 days of the grades being entered. No changes will be made after this point.**

Other Notes

- Form teams of 2-3 for the presentations in class and the in-class exercises
- In-class exercises
  - A typical lecture will start with covering a new topic followed by a hands-on in-class exercise in which you need to code and submit your work at the end for bonus points.

- For in-class presentations:
  - Send me an email with the names, emails, and WSU IDs of the members of the team (cc all team members) and the presentation that you are interested in.
  - If you don't have a team send me the info for yourself and you will be matched with others interested in the same presentation.
  - If you don't have preference on the topic that's fine too – I will assign you to whatever is available.
  - There are two deliverables for this – check the schedule for the deadlines.

- Things might feel a little slow in the beginning but do NOT get used to this pace. We will accelerate significantly.
  - Homework assignments will be definitely getting longer than homework 1

- Ask questions, keep up with things, work hard and the course shouldn't be that difficult. If you do not do these things then the course may end up being excessively difficult.

In class

- I will ask you to fill short surveys to touch base with you, to get quick input, and to keep track of your participation in class. Those will be through Google Forms so please create a Google account if you don't have one.

- Links to those surveys will be posted on Canvas (check the Surveys tab during class time).

- **Surveys will be available during class time!**

First short survey

- Open your web browser
- Go to Canvas page for the course -> Modules -> Surveys
- Fill in the survey for today

TODOs before next class

1. **TODOs on Canvas**
   - Update your profile picture (to help me remember your face) and your bio (to help me to get to know you better)
   - Subscribe to all forums in the discussion board and the announcements. (Otherwise, you might miss announcements!)
   - Read the syllabus carefully!

2. **Setup your working environment for the HWs and the in-class exercises**
   - Install VS (if you are planning to build a **Windows** Desktop Application)
     OR
   - Install JetBrains Rider (if you are planning to build a **cross-platform** Desktop Application)

3. **Check that your EECS GitLab account is active.** You should do this periodically (1-2 days before every HW is due) – you will lose access if your password expires.