Intro to the Arduino IDE, the Teensy 3.2, & ARM Technology
Hardware, Software & Resources

John R. Wright, Jr. PhD, CSTM
ITEC 467, Mobile Robotics
IDE = Integrated Development Environment

- An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software.

- Typically, an IDE contains a code editor, a compiler or interpreter and a debugger that the developer accesses through a single graphical user interface (GUI).

- An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications.

  http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment

- JAVA uses Eclipse as its IDE

- We use C++ to Program the Teensy
  - via a patch called “Teensyduino”
  - utilizing the Arduino IDE

  https://www.pjrc.com/teensy/teensyduino.html
Teensy 3.2

- 32 bit ARM processor
  - ARM = Advanced RISC Machine
    - RISC = Reduced Instruction Set Computer
- 72MHz Cortex-M4 Technology
Intro to ARM

- Watch Video (44:25):
  http://whatis.techtarget.com/definition/ARM-processor

Cortex-A

Highest performance
Optimized for rich operating systems

Learn more about the Cortex-A series processors

Cortex-R

Fast response
Optimized for high-performance, hard real-time applications

Learn more about the Cortex-R series processors

Cortex-M

Smallest/lowest power
Optimized for discrete processing and microcontroller

Learn more about the Cortex-M series processors

SecurCore

Tamper resistant
Optimized for security applications

Learn more about SecurCore processors
The Arduino IDE C++ Language/Environment General Videos

- Tutorial 03: Arduino IDE and Sketch Overview (8:26)
  https://www.youtube.com/watch?v=YDkdVZ7e3OY
- Tutorial 04: Understanding Arduino Syntax (10:22)
  https://www.youtube.com/watch?v=AJkEO9t4WfY
- Tutorial 05: Understanding Variables (11:24)
  https://www.youtube.com/watch?v=QNTaO5qjniE&t=557s
- Tutorial 06: Blink an LED (14:38)
  https://www.youtube.com/watch?v=33sNhlekRV8
- Tutorial 07: digitalRead() and the Serial Port (21:21)
  https://www.youtube.com/watch?v=ClrN2CaO6bo
- Tutorial 08: analogRead() and the Serial Port (12:02)
  https://www.youtube.com/watch?v=CPMQSXrsaO8
- Tutorial 11: If Statement Conditionals (14:12)
  https://www.youtube.com/watch?v=w-yfj1VWXW8I
- Tutorial 12: For Loop Iteration (22:33)
  https://www.youtube.com/watch?v=bMsT8FeMQ-k

Other videos exist and are helpful, but we are using the Teensy Microcontroller and a different carrier board so you might just want to refer to Patton Robotics Videos from here or use the Odom textbook(s).
Getting Started (Setting Up)!

Step 1 – Download Arduino 1.8.1

Step 2 – Download Teensyduino 1.35

Follow instructions and make sure this is installed in same location as the Arduino software.

Step 3 – Open the Arduino program once and then close it.

Step 4 – Open Arduino software, connect microcontroller to USB of computer

Step 5 – Under Tools…Board Menu select the microcontroller you are using

Step 6 – Write a short program and try to connect. You may need to be patient the first time as it takes a little time the very first time we launch the Teensyduino.
Follow along with Brian Patton at Patton Robotics (14:36):

- https://www.youtube.com/watch?v=11Xo-WO-2S0&list=PLYI2Xb6BPCrrspYlBxrcnIlo83AgmQy22&index=1

Brian is the inventor of the Teensy Motherboard (Carrier Board) that we use and used to be with RoboOdyssey – maker of the BX24p (BasicX) microcontroller.

- Be careful when using the microcontroller without the carrier board – pins might short out the device if they should touch metal (use a wood table or use on top of a piece of paper)!
PRT Motherboard (Carrier Board)

- PRT Motherboard tutorial 1 (12:08)
  - Intro to hardware and USB Voltage Isolator
  - https://www.youtube.com/watch?v=9in-VuT08g0&index=10&list=PLYl2Xb6BPCrriVQfwxfCJdSLrF6Chkh6R&t=73s
Odom’s Text - Video Tutorials

- Tutorials (Playlist) that coincide with your Vol 1 text:
  - https://www.youtube.com/playlist?list=PLYI2Xb6BPCrqQGXBvGjTTHlao0WinglVoc

- Use as needed as you develop your robot (i.e. math functions, variable types, etc.)
Responsibilities (Senior Capstone R&D)

- To learn the code on your own using the provided resources. You will likely learn as you go – as you need the capability. There are multiple ways to approach the maze. Each group will have a different approach. Teaching will be done side-by-side as a result. You must research the code, however.

- You are responsible for the technology you check out! Replacement sensors and microcontrollers can be found here:
  - www.pattonrobotics.com
  - https://www.pjrc.com/teensy/

- You are expected to spend a considerable amount of time outside of class working on this R&D project. 100 hours of outside class development time is typical per robot/team!