

Forward Kinematics

Design Problem 1

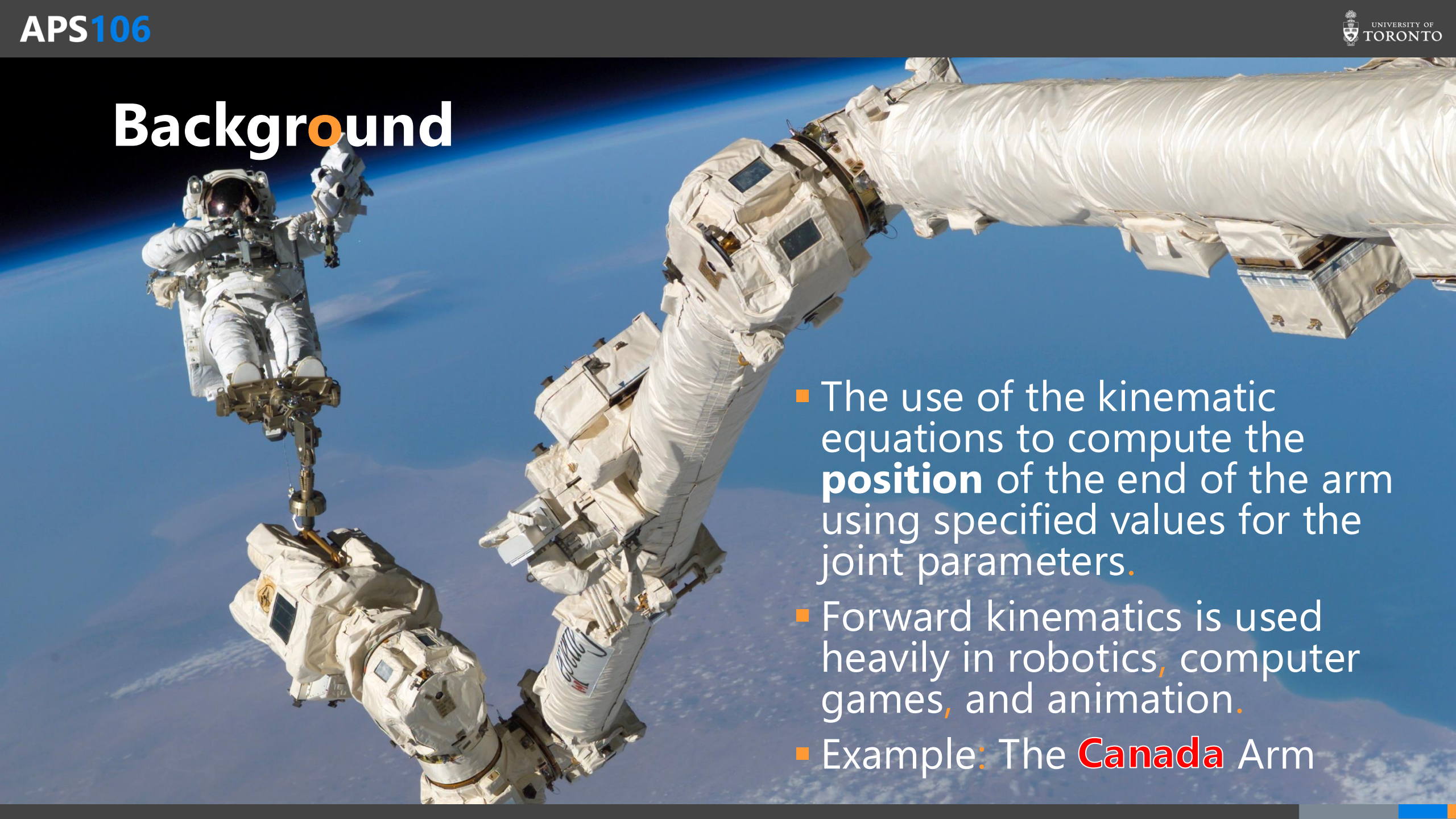
if nothing else, write `#cleancode`

Agenda

- Problem Background
- Learning Objectives
- Live Coding!

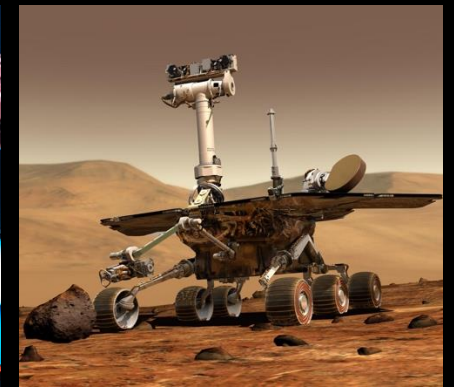
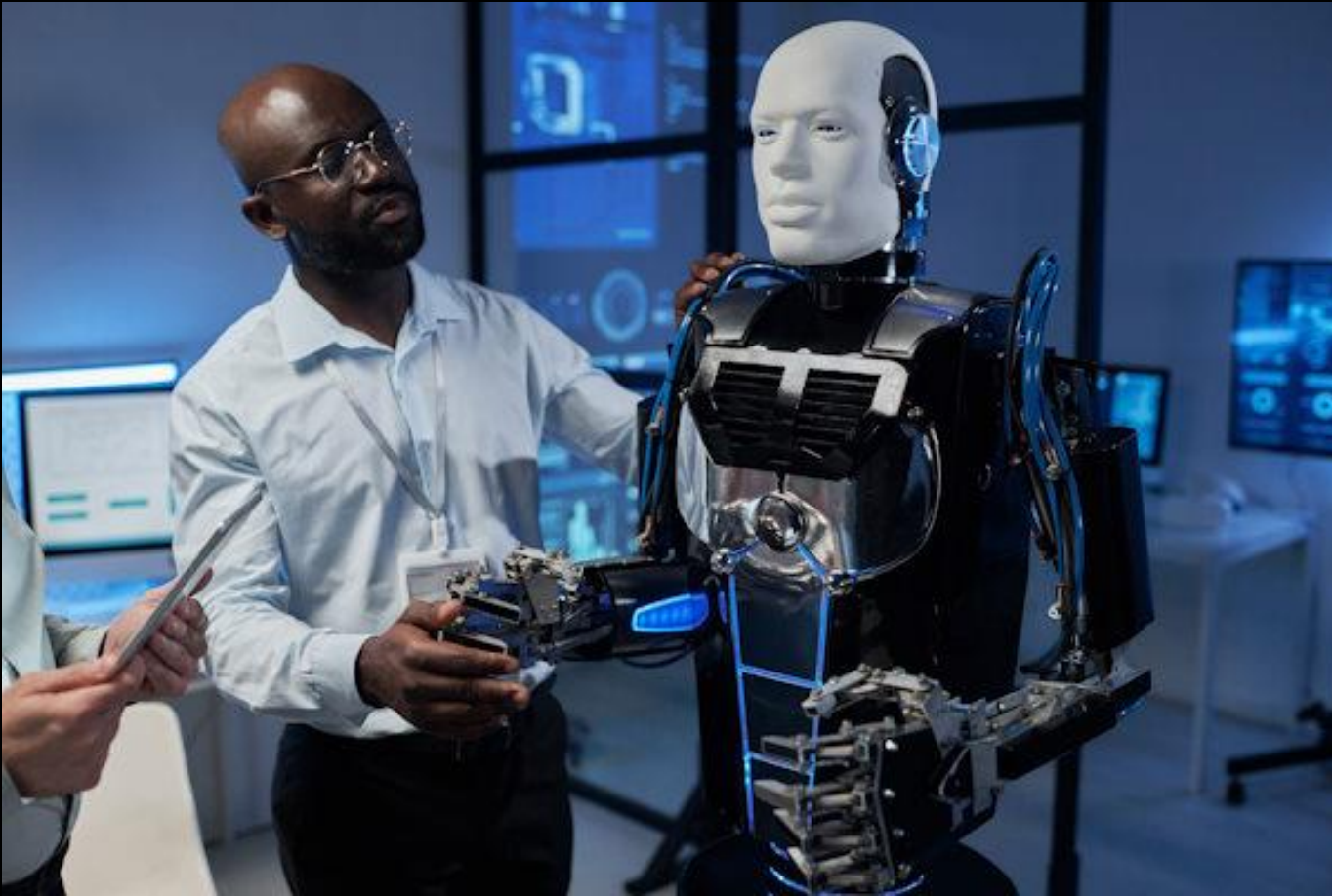


Background

- 
- The background image shows a full-body view of an astronaut in a white spacesuit floating in space. The astronaut is positioned on the left side of the frame, facing towards the right. They are holding onto a complex mechanical structure, which is part of the International Space Station. This structure includes a large, white, cylindrical component that resembles a robotic arm, extending from the right side of the frame towards the center. The arm is covered in white thermal blankets and has various sensors and instruments attached to it. The background is a clear blue sky with a thin layer of white clouds visible near the horizon.
- The use of the kinematic equations to compute the **position** of the end of the arm using specified values for the joint parameters.
 - Forward kinematics is used heavily in robotics, computer games, and animation.
 - Example: The **Canada** Arm

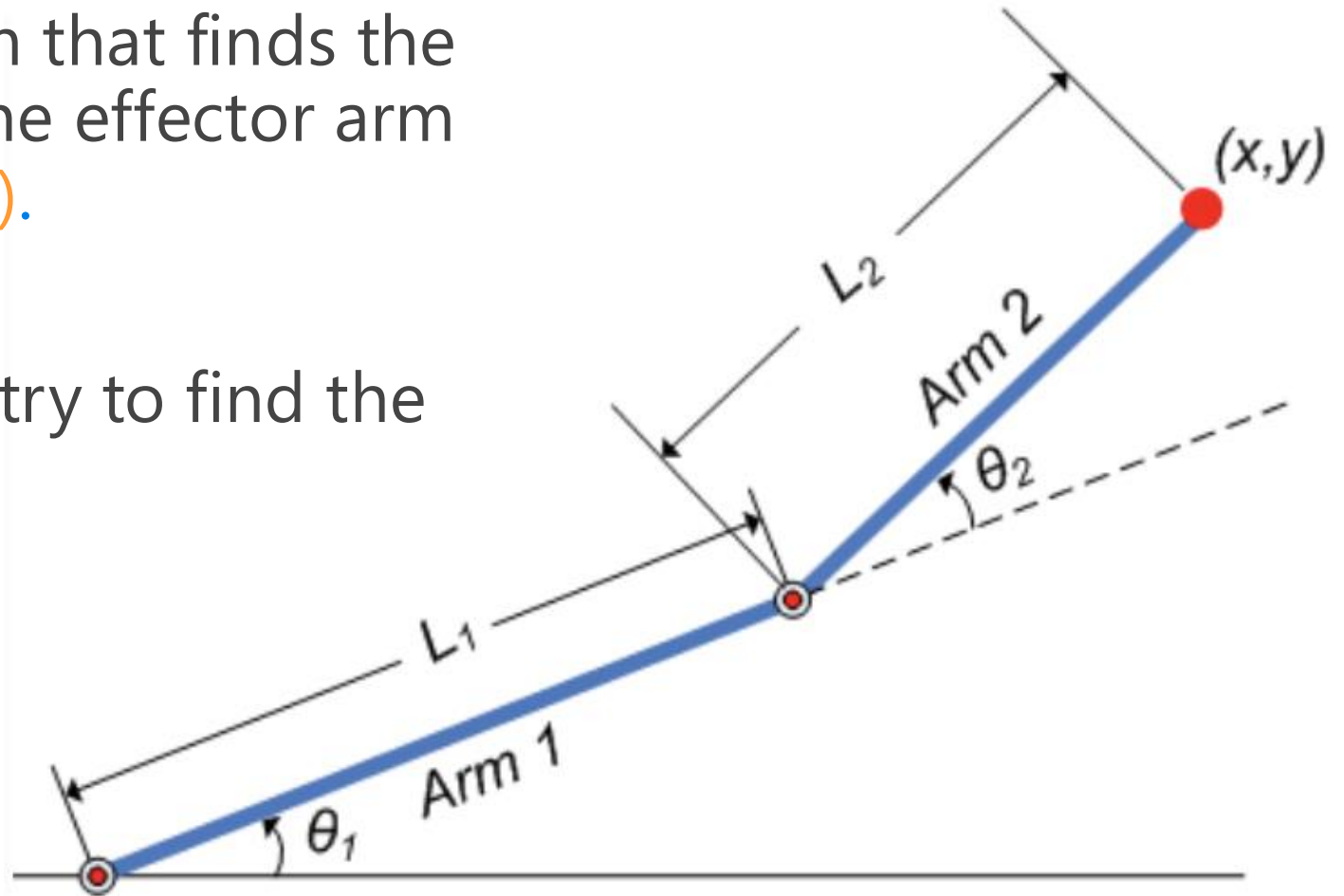
Other Cool Examples:

Humanoid and Industrial Robots, Animations and Gaming, 3D printing, ...



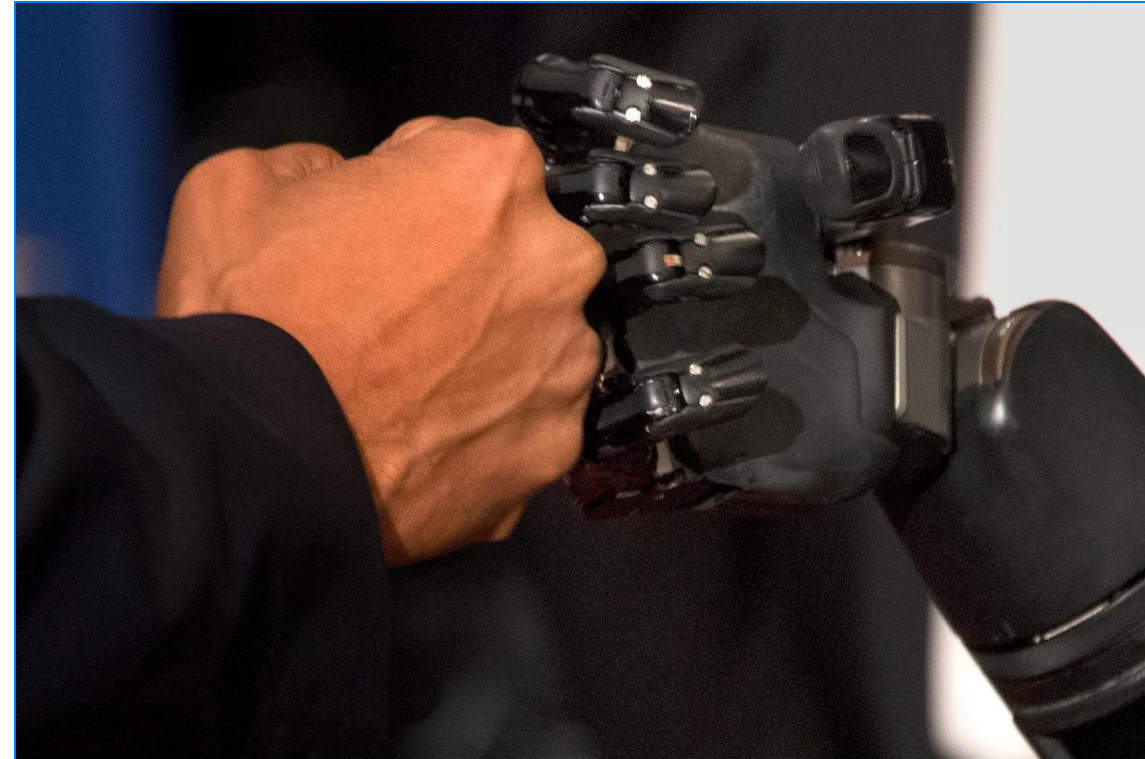
Background

- Goal: Create a program that finds the x- and y- position of the effector arm (i.e. the end of Arm #2).
- Will rely on trigonometry to find the positions.



Learning Objectives

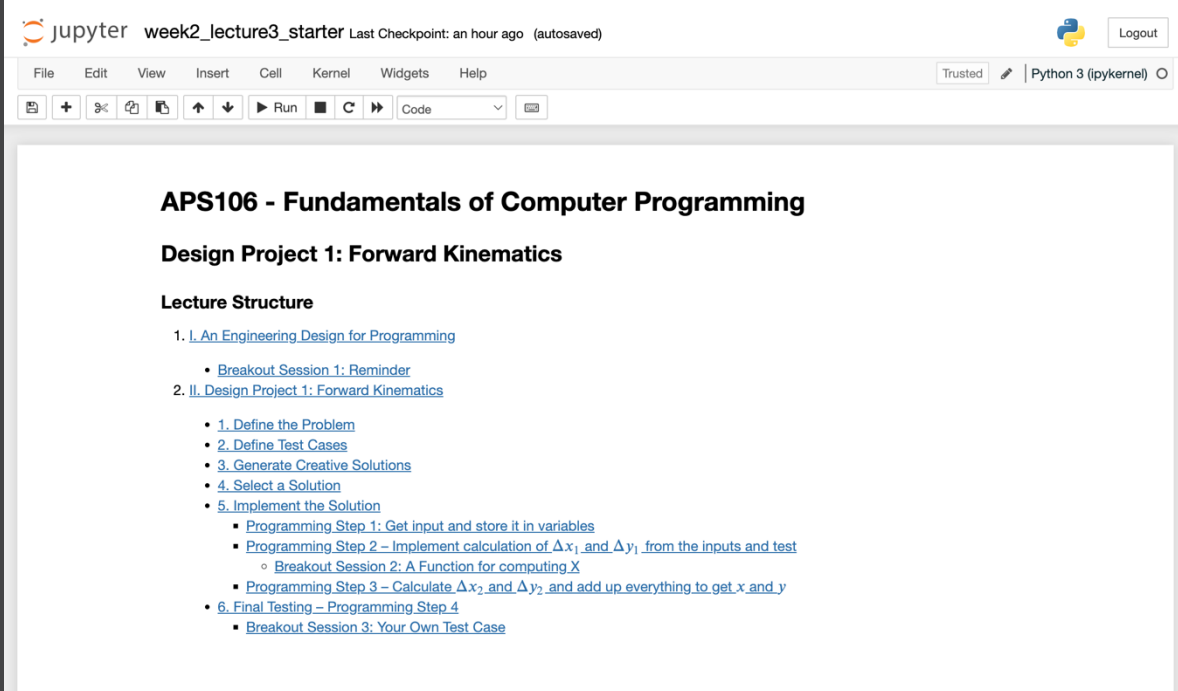
- Practice with user input.
- Learn to define and use custom functions.
- Practice with built-in libraries.



Forward Kinematics

Design Problem 1

Let's review the notebook now!



jupyter week2_lecture3_starter Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

APS106 - Fundamentals of Computer Programming

Design Project 1: Forward Kinematics

Lecture Structure

1. [I. An Engineering Design for Programming](#)
 - [Breakout Session 1: Reminder](#)
2. [II. Design Project 1: Forward Kinematics](#)
 - [1. Define the Problem](#)
 - [2. Define Test Cases](#)
 - [3. Generate Creative Solutions](#)
 - [4. Select a Solution](#)
 - [5. Implement the Solution](#)
 - [Programming Step 1: Get input and store it in variables](#)
 - [Programming Step 2 – Implement calculation of \$\Delta x_1\$ and \$\Delta y_1\$ from the inputs and test](#)
 - [Breakout Session 2: A Function for computing X](#)
 - [Programming Step 3 – Calculate \$\Delta x_2\$ and \$\Delta y_2\$ and add up everything to get x and y](#)
 - [6. Final Testing – Programming Step 4](#)
 - [Breakout Session 3: Your Own Test Case](#)

if nothing else, write `#cleancode`