



Lecture 26: Nothing is as Exciting as Civic Duty and - oh yeah - Gambling!  
*(voting example (cont.) and playing cards example)*



# Announcements and reminders

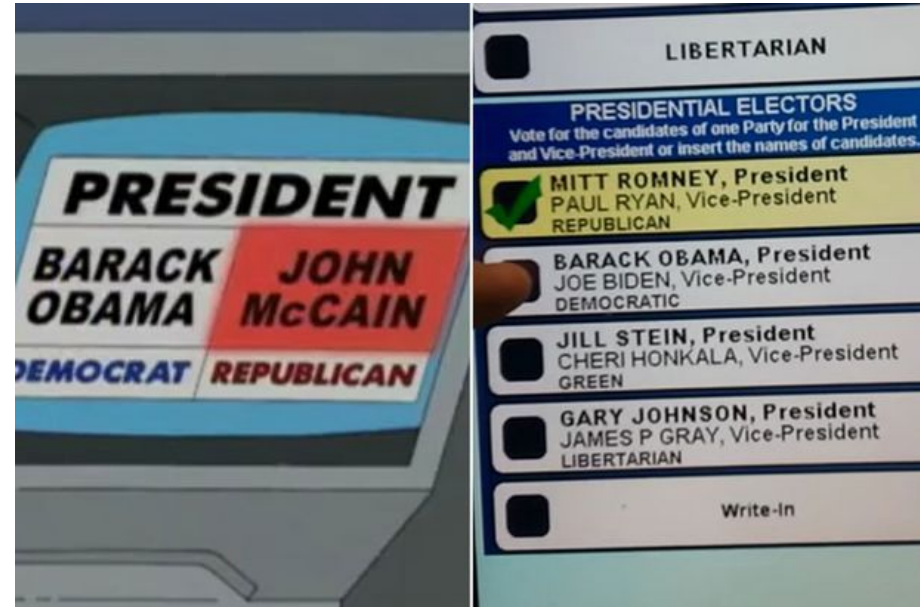
---

- Project 2 posted, due **Saturday March 23 by 6 PM**  
(no early bonus aside from being a badass)
- Practicum 2 practice problems up ← **do them**
  - Both **MCQ + Coderunner**
  - Same rules as Practicum 1  
(cheat sheet, Cloud9, etc...)
  - Let your TA know about any conflicts, and **include documentation**



# Let's simulate an election!

**Example:** Write code to simulate an election.

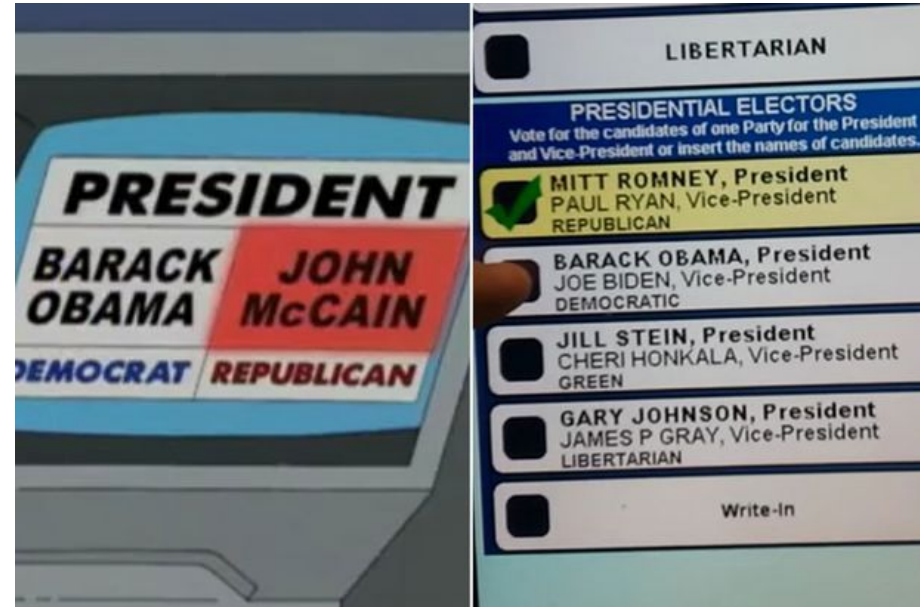


# Let's simulate an election!

**Example:** Write code to simulate an election.

Wait... that's it? Okay....

What do we need to get started?



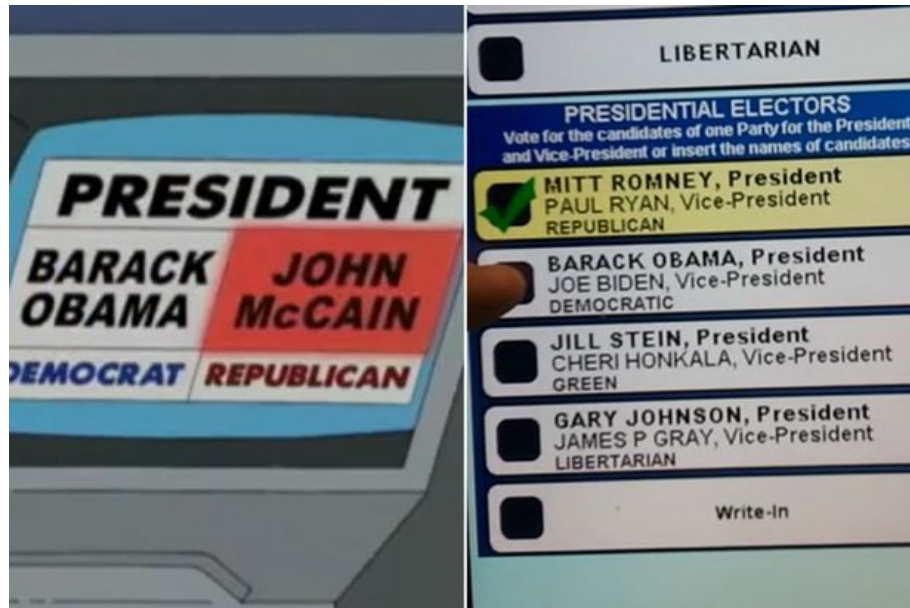
# Let's simulate an election!

**Example:** Write code to simulate an election.

Wait... that's it? Okay....

What do we need to get started?

- Voting machines
- Voting center
- Probabilistic simulation of voter choices



# Let's simulate an election!

**Example:** Write code to simulate an election.

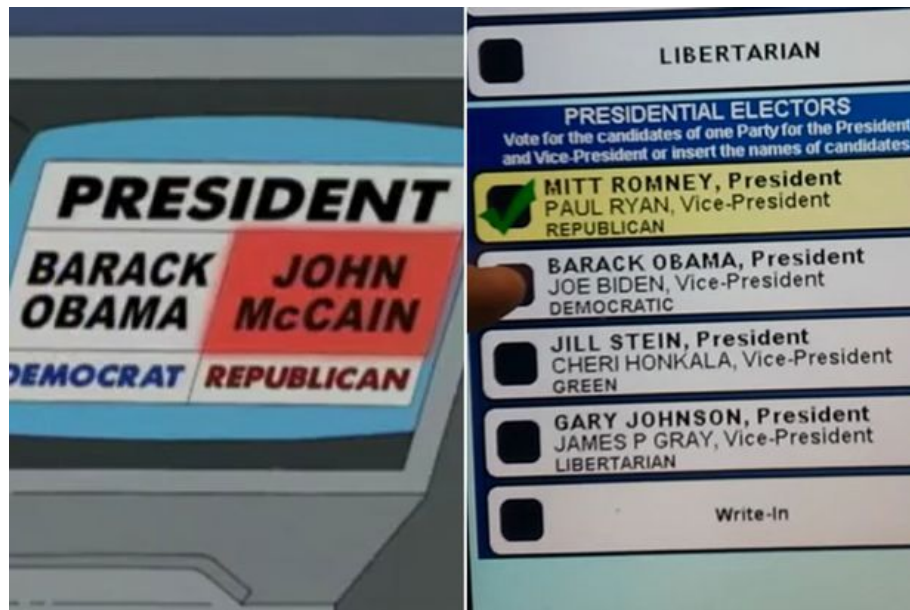
Wait... that's it? Okay....

What do we need to get started?

- Voting machines (**VotingMachine.h**)
- Voting center (**VotingCenter.h**)
- Probabilistic simulation of voter choices  
(**testVotingCenter/Machine.cpp**)

→ Classes for voting machines and centers

→ rand() simulation for who votes for what





# Let's simulate a deck of cards!

---

**Example:** Write code to simulate a deck of cards.

What do we need to get started?



# Let's simulate a deck of cards!

---

**Example:** Write code to simulate a deck of cards.

What do we need to get started?

- Start with a class for individual **Cards**
  - Suit and number
- Then make a class for the whole **Deck**
- **Constructors** -- default and parameterized?

And - of course! - a driver function to test all our goodies.

→ *Card.cpp, Deck.cpp, deckDriver.cpp*





