

# CPSC 380

---

Joseph Hansen

[jhansen@chapman.edu](mailto:jhansen@chapman.edu)

# PROJECT 2

## SEE-SAW SIMULATOR

---

# Project Description

- You are building a see-saw simulator that emulates two individuals that weigh the same going up-and-down. On one end you have person A, Fred. Whenever he pushes up, he creates an effective upward velocity of 1 ft/sec. On the other end you have person B, Wilma. Whenever she pushes up she creates an effective upward velocity of 1.5 ft/sec. Whenever, person A or B comes within 1 ft of the ground they will push back up causing the opposing person to come down. The maximum height any person will go is 7 feet. Using two threads and two semaphores, write a simulator that emulates these conditions

# Hints

- Assume you have a main thread that starts the two-threaded see-saw – it is not part of the simulator
  - Once the simulator completes the threads should join the main one
- Assume at time 0 person A is at the low end, one foot from the ground ready to push up. This means person B is at the high end, 7 feet off the ground.
- At the 1 second mark, person A is 2 feet off the ground and person B is 6 feet off the ground. They continue to travel in opposite directions until person B is 1 foot off the ground.
- When person B is 1 foot off the ground, she will push upwards and reverse direction. Remember Wilma is stronger than Fred, so one second later she will be 2.5 feet off the ground and Fred will be 5.5 feet.
- Assume simulator once started will run 10 times (up-and-down)
- In order to throttle each process, you can use a sleep (1 second)
- Remember you are trying to synchronize the two threads

# Threads

- `void *fredSee( ) //fill in Fred's behavior`
- `{`
- 
- `.....`
- 
- `}`
- 
- `void *wilmaSaw( ) //fill in Wilma's behavior`
- `{`
- 
- 
- `....`
- 
- `}`

# Project Artifacts

- Demonstrate by outputting the height of each individual once per second
  - Can be printed to the screen or sent to a file
- Students must turn in the following:
  - Source code
  - Executable
  - Output of the program
  - A brief design description that explains the usage and type of semaphores in establishing the required synchronization