Carlo Ace Sagad CS340

**Project 2**

Binary Semaphore mutex = 1

Binary Semaphore forest = 1

Binary Semaphore lake = 0

Counting Semaphore field = 0

Counting Semaphore judge = 0

Counting Semaphore teamRed = 0

Counting Semaphore teamGreen = 0

athletesInCafeteria = 0

disqualifiedAthletes = 0

athletesRed = 0

athletes\_count = 0

Athlete()

sleep(random\_sleep\_time) // Simulate rest or food before the first obstacle

P(forest)

P(mutex)

athletes\_count++

V(mutex)

forest\_search(priority)

P(mutex)

athletes\_count--

if athletes\_count == 0 V(lake);

V(mutex)

V(forest)

P(lake)

P(teamRed) // Wait for teamRed semaphore

sleep(random\_sleep\_time)

atheletesRed++

V(lake)

// Rest after completing the lake obstacle

sleep(random\_sleep\_time)

V(teamRed) // Release teamRed semaphore

P(lake)

P(mutex)

athletesRed--

if athletesRed == 0:

V(judge) // Wake up the judge when all athletes are ready for the next obstacle

V(mutex)

P(mutex)

athletesInCafeteria++

if athletesInCafeteria == num\_athletes\_per\_team \* num\_teams:

V(judge) // Wake up the judge when all athletes are in the cafeteria

V(mutex)

P(judge) // Wait for the judge to announce the start of the race

// Simulate the race

sleep(random\_sleep\_time)

// Announce the completion of the race

P(mutex)

if not disqualified:

// Display results and awards

V(mutex)

Judge():

sleep(random\_sleep\_time) // Simulate commute time to the cafeteria

P(mutex)

athletesInCafeteria++

V(mutex)

if athletesInCafeteria == num\_athletes\_per\_team \* num\_teams:

V(judgeSemaphore) // Signal athletes that all are in the cafeteria

sleep(random\_sleep\_time)

// Announce the start of the race

P(mutex)

V(judgeSemaphore) // Signal athletes to start the race

V(mutex)

else:

V(mutex)

P(judgeSemaphore) // Wait for athletes to arrive in the cafeteria

// Compile points and times, decide winning team by tossing a coin

sleep(random\_sleep\_time)

// Signal the winning team

V(teamSemaphore)

sleep(random\_sleep\_time)

// Signal the losing team to leave

V(teamSemaphore)

sleep(random\_sleep\_time)

P(mutex)

athletesInCafeteria--

if athletesInCafeteria == 0:

V(judgeSemaphore) // Signal that all athletes have left

V(mutex)