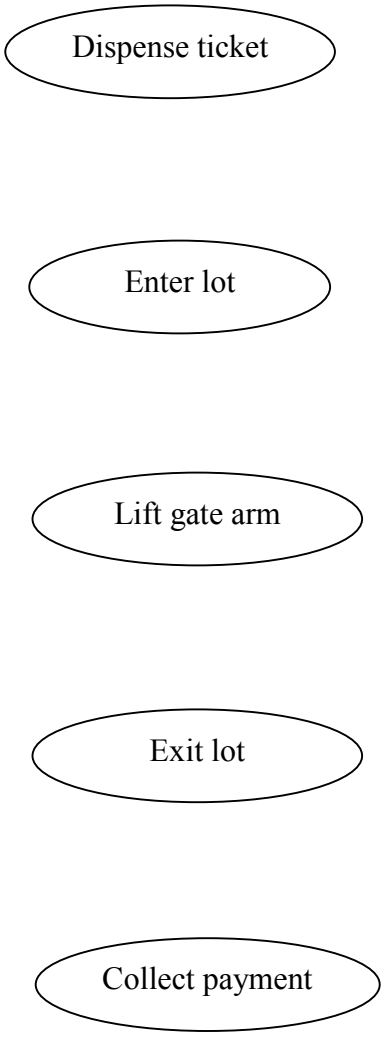
Sample Final Exam

1. Given the specifications listed below, draw a **Use Case Diagram** for an **Automated Parking Lot (APL) Controller**. Some of the use cases are included for you. (10 points each)

a. There is one entry gate to the parking lot, which interfaces to the APL controller through a **Push-for-ticket button**, a **Ticket dispenser**, and the **Entry gate arm**. The **driver** initiates the *Enter lot* use case by pushing the **Push-for-ticket** button. The *Enter lot use* case then starts the *Dispense ticket* use case, which sends a command to the **Ticket dispenser**. After the driver removes the ticket, the *Enter Lo*t use case starts the *Lift arm* use case, which sends a command to the **Entry gate arm**.

b. There is also one exit gate, which interfaces to the APL controller through a **Ticket reader**, a **Credit card reade**r, and the **Exit gate arm**. The **driver** initiates the *Exit lot* use case by inserting the ticket into the **Ticker reader**. The *Exit lot* use case then computes the parking fee and starts the *Collect payment* use case, which uses the **Credit card reader** to read the card and then transfers the payment via an external **Credit card banking system**. When the payment has been collected, the *Exit lot* use case starts the *Lift gate* use case to lift the **Exit gate arm**.

c. If an error occurs in the operation of dispensing the ticket at the entry gate, the **Lot operator** intervenes with manual control, for example, the dispenser may be out of tickets or it may have jammed. Also, if an error occurs in the operation of the gate arm at the entry or exit gates, the **Lot operator** can intervene by pushing a manual button. Show the handling of these error conditions with use cases.



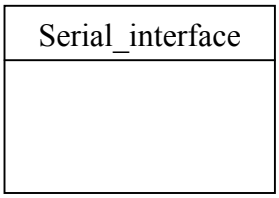
Sample Final Exam

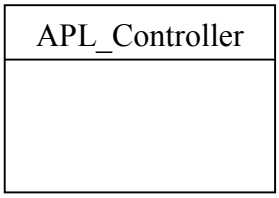
2. Complete a **UML Class Diagram** of the **Automated Parking Lot Controller** according to the specifications given. Include the multiplicity on your diagram, assuming 1-2 entry gates and 1-2 exit gates. (10 points each)

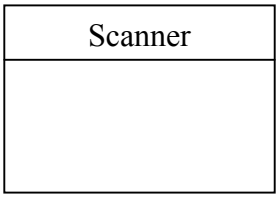
a. The **Ticket_button_interface** and the **Ticket_dispenser_interface** are both specializations of the **Serial_interface** class. Both the Ticket_button and Ticket_dispenser interfaces are components of the **APL_controller**.

b. The **Ticket_reader** and the **Credit_card_reader** are both specializations of the **Scanner** class. Both of these readers are components of the **APL_controller**.

c. The **APL_controller** also contains a **Gate_arm_controller** and a **Banking_system_interface**. The **APL_Controller** is responsible for creating and deleting any **Gate_arm_controller** objects; a **Gate_arm_controller** object cannot exist independently from the **APL_Controller** object.







Sample Final Exam

3. Draw a **UML Statechart** of an **Automated Car Wash System** given the specifications listed below. Note that there are two parallel lanes that can be in operation at the same time (on different cars). One lane is called the **Economy Lane** and the second is the **Full Service Lane**. State any assumptions you have made. (10 points each)

a. The **Economy Lane** transitions through a set sequence, based on time. As the car drives into the lane, a sensor detects the presence of the car in the starting location, and the *ev_start_wash* begins the process. The car always starts out in the **Rinse** state for *10 seconds*, then moves to the **Soap** state for *15 seconds*, then on to the **Wash** state for *30 seconds*. When finished with the **Wash** state, the **Rinse** state starts again for *30 seconds*, after which the car wash is complete.

b. The **Full Service Lane** offers two types of services. When the car is in position, the driver presses the Premium button, and the resulting *ev_prem_button* begins the Premium wash cycle after waiting *10 seconds* (for the driver to raise the window). In this cycle, the car first enters the **Rinse** state for *20 seconds*, then moves to the **Soap** state for *20 seconds*, then on to the **Wash** state for *40 seconds*. When finished with the **Wash** state, the **Rinse** state starts again for *40 seconds*, and then the car enters the **Dry** state for *20 seconds*. After this, the wash cycle is complete.

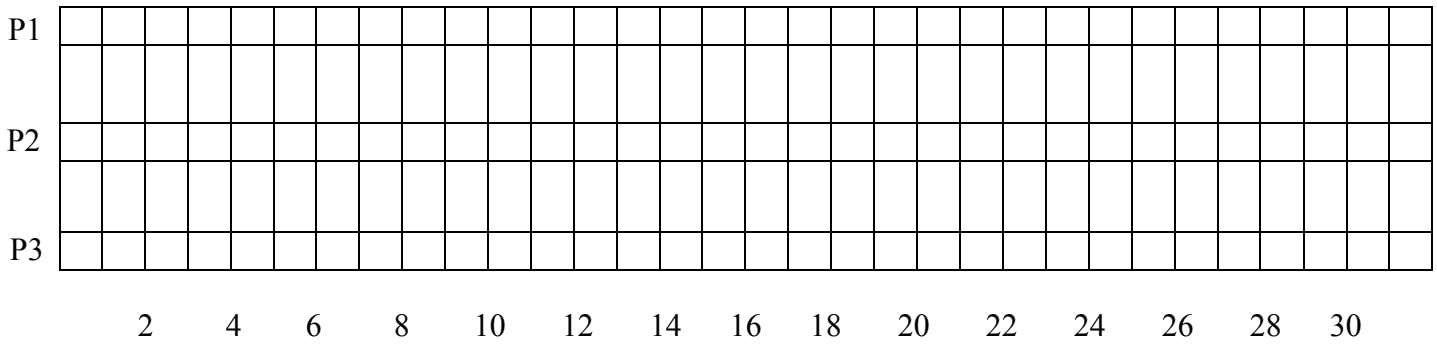
c. The second type of service in the **Full Service Lane** is the Luxury wash cycle. When the car is in position, the driver presses the Luxury button, and the resulting *ev_lux_button* begins the Luxury wash cycle after waiting *10 seconds*. In this cycle, the car first enters the **Rinse** state for *20 seconds*, then moves to the **Soap** state for *20 seconds*, then on to the **Wash** state for *40 seconds*. When finished with the **Wash** state, the **Wax** state runs for 20 seconds, then the **Rinse** state starts again for *40 seconds*, and then the car enters the **Dry** state for *20 seconds*. After this, the wash cycle is complete.

Sample Final Exam

4. Consider 3 processes: P1 = (2, 12, 12) P2 = (3, 8, 8) P3 = (2, 9, 9).

a. Verify the feasibility of Rate Monotonic scheduling. Justify your result. (10 points)

b. List the process priorities and construct the schedule using Rate Monotonic scheduling, taking the schedule to time unit 32. (10 points)

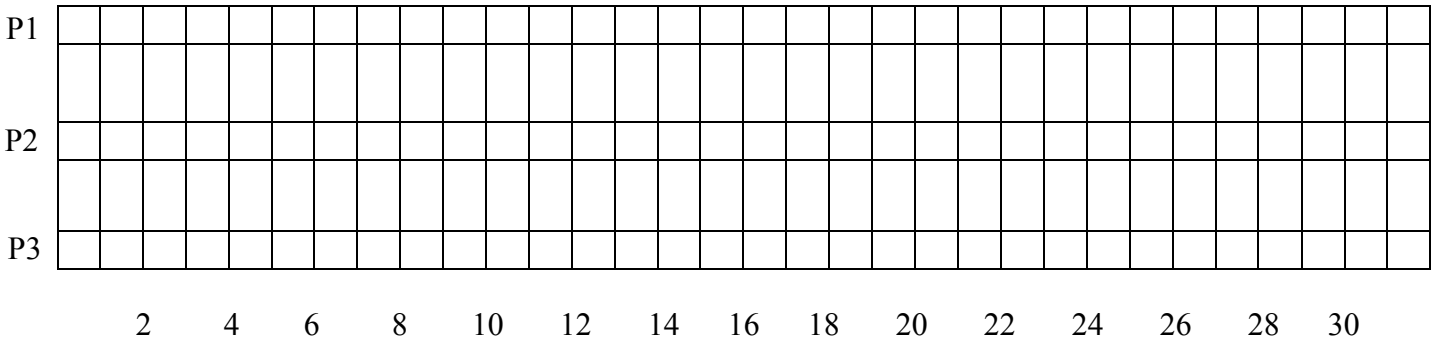


Sample Final Exam

5. Consider 3 processes: P1 = (5, 12, 12) P2 = (4, 10, 10) P3 = (2, 18, 18).

a. Verify the feasibility of Earliest Deadline First scheduling. Justify your result. (10 points)

b. Construct the schedule using Earliest Deadline First scheduling, taking the schedule to time unit 32. (15 points)



Sample Final Exam

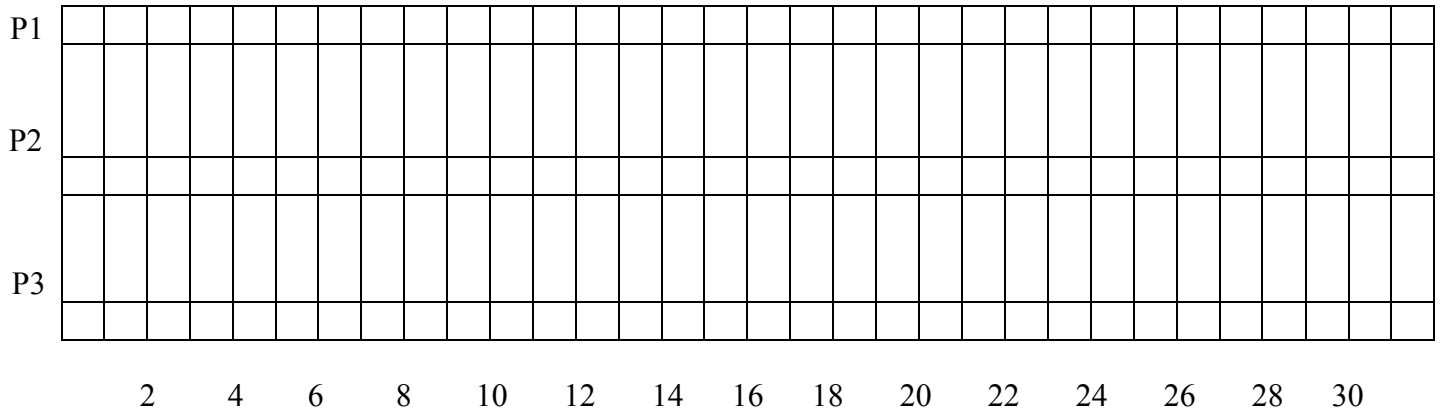
6. Consider the 3 processes below. Assume that $\pi_{P1} > \pi_{P2} > \pi_{P3}$

P1:: begin … lock(S1); CSa; lock(S3); CSb; unlock(S3); CSc; unlock(S1); end;

P2:: begin … lock(S3); CSd; lock(S2); Cse; unlock(S2); CSf; unlock(S3); end;

P3:: begin … lock(S2); CSg; lock(S1); CSh; unlock(S1); CSi; unlock(S2); end;

a. Show an execution sequence that leads to deadlock, assuming no priority inversion protocol. (10 points)



b. Show how the priority ceiling protocol prevents deadlock on your sequence above. Include the priority ceilings of the semaphores. (15 points)