CS132: Software Engineering

Midterm Exam

13:00-14:40, June 10th, 2020

There are 5 problem sets and the total points are 80 points. Your score will be divided by 4 to reflect the 20% constitution for the midterm. Each problem set includes a few questions. For each question, the maximum possible points are stated.

Please write your answers **legibly on the answer booklet** so that we can read and understand your answers. If a problem seems ambiguous, please feel free to state your assumption explicitly and solve the problem. Obviously, your assumption should be reasonable and should not trivialize the problem.

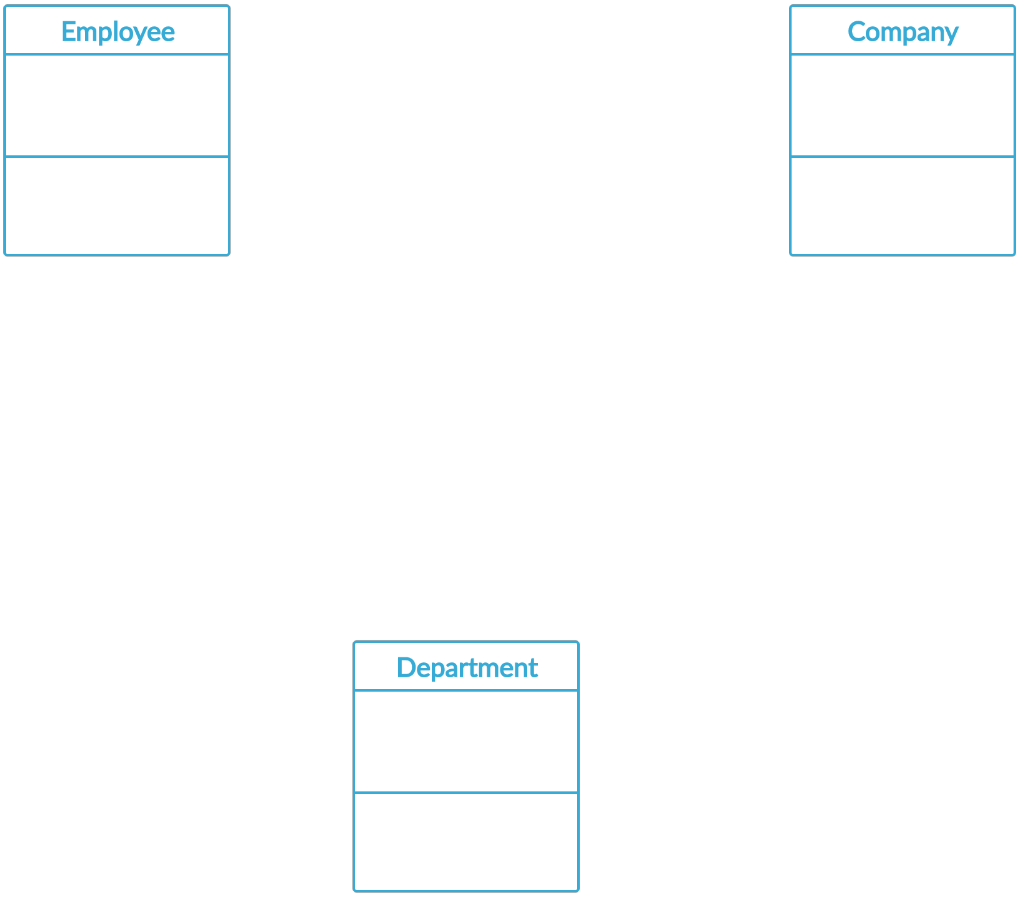
Pledge. Copy the following pledge and sign your name in your answer booklet:

I neither cheated myself nor helped anyone cheat on this exam.

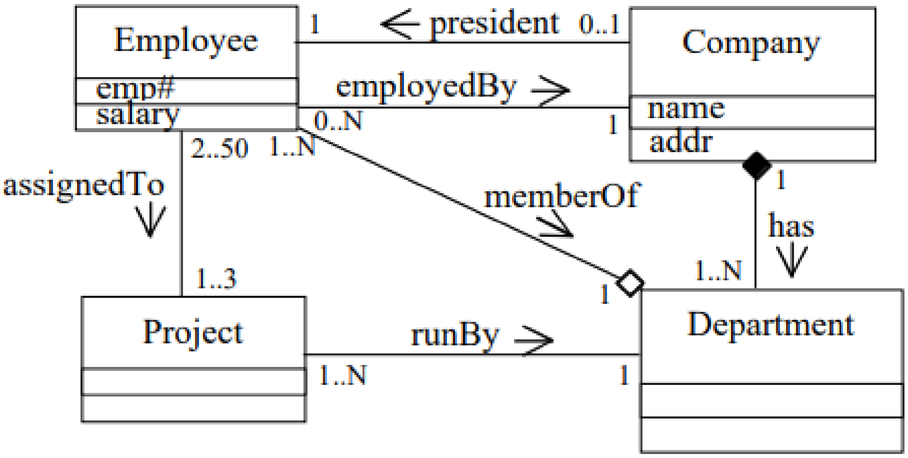
Problem 1. (20 pts)

Companies employ employees (who can only work for one company), and consist of one or more departments. Each company has a single president, who is an employee. Departments have employees as members and run projects (one or more.) Employees can work in 1 to 3 projects, while a project can have 2 to 50 assigned employees. You may assume that companies have a name and address, while employees have an ID and a salary.

A partial class diagram is given to you below. Please complete the class diagram. Make sure to show attributes, multiplicities and aggregation associations, where appropriate. No need to show any operations.



Solution:



Problem 2. (20 pts)

Consider the following C program snippet. All variables are integers. Recall that the integer division in C causes roundings, e.g., 7/3=2, and 8/3=2.

```
if (a > b && x == 6) {
    x = x / a;
    a = a - b;
  }
if (a == 2 || x > 1) {
    x = x + 1;
  }
```

1. What is the minimum number of test cases for **branch coverage** testing? (2pts)
2. Please specify a test suite that achieves **branch coverage** with the minimum number of test cases. (8pts)
3. What is the minimum number of test cases for **condition coverage** testing? (2pts)
4. Please specify a test suite that achieves **condition coverage** with the minimum number of test cases. (8pts)

(a) Two test cases are needed. One can use a=3; b=1; x=6 to make both branches true, and then use a=3; b=4; x=0 to make both branches false. (b) Two test cases are needed. One can use a=3; b=1; x=6 to make all four conditions true, and then use a=3; b=4; x=0 to make all four conditions false.

Problem 3. (20 pts)

Read the description below carefully and complete the UPPAAL templates, which can be assembled into a model of arcade game machine.

An arcade machine consists two components: the game host machine and a coin acceptor module. Assume you are given the following code in global declaration section:

int coins = 0; // Denote the current number of coins inserted to the machine.
bool power = false; // Denote whether the machine is powered. "*insert*" will be fired even if the machine has no power because its a physical signal!

There are some broadcast channels already defined for you. Assume they are all physical signals and automatically fire in condition.
broadcast chan insert; // It automatically fires when a coin is physically inserted.
broadcast chan boot; // It automatically fires when the machine is connected to power.
broadcast chan shut; // It automatically fires when the machine shuts down;
broadcast chan start; // It automatically fires when start button on the machine is pressed.
broadcast chan refund; // It automatically fires when refund lever is pulled.

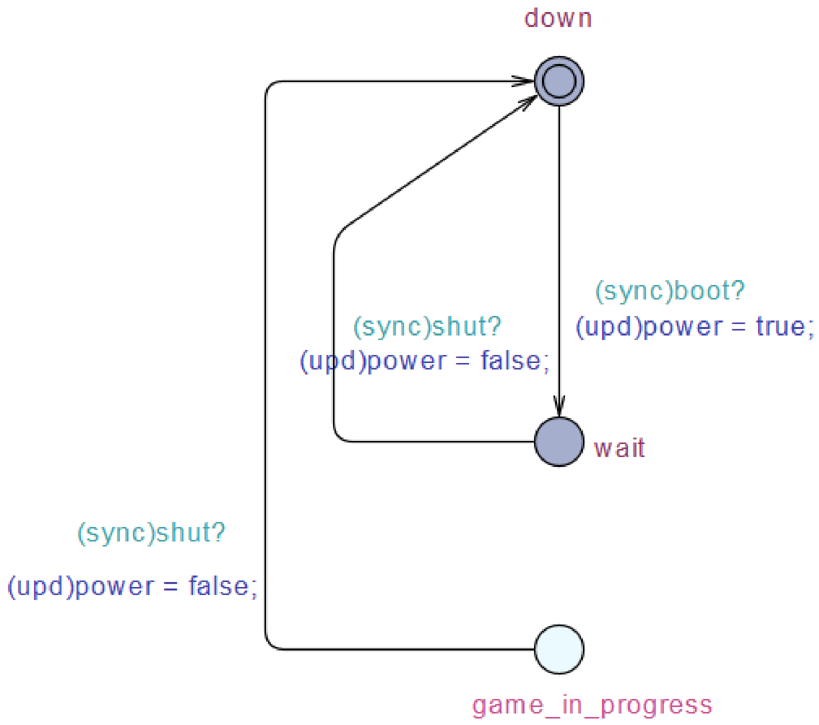
Please label the type of statement you are making on the transitions. For example, if you want to add a guard to the transition, you need to write down "(guard) t < 3". Use "sel" and "upd" to represent selections and updates, and "inv" to denote invariants.

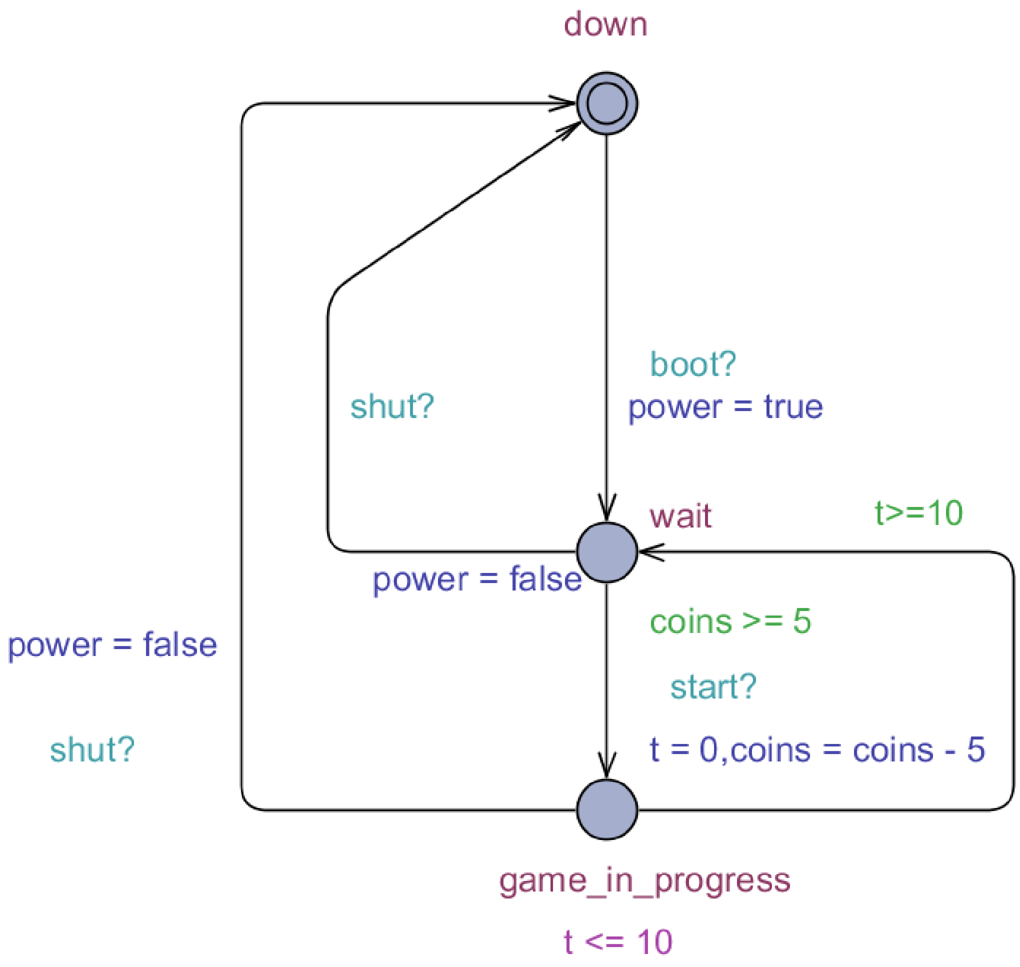
1) Complete the UPPAAL template of the game host machine base on the following information: (6pts)

1. The game requires 5 coins to start. i.e. each game start will consume 5 coins.
2. The game will start if therere 5 or more coins inside when the start button is pressed.
3. The game will run 10 minutes once start unless a forced shutdown happens.
4. A timer, t, is defined in this template. The unit is minute.

Note:

- You are allowed to add urgent/committed locations, but not regular locations.
- You are allowed to add edges or add statements on existing edges.





2) Complete the UPPAAL template of the coin acceptor. (10pts)

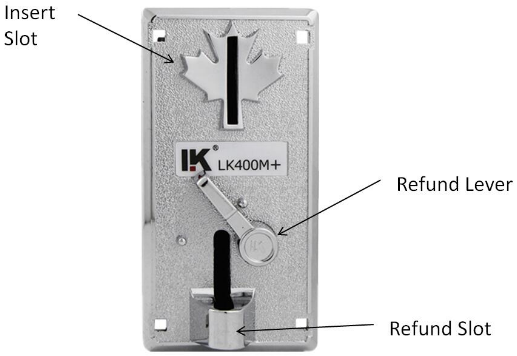
A typical arcade coin acceptor looks like this:

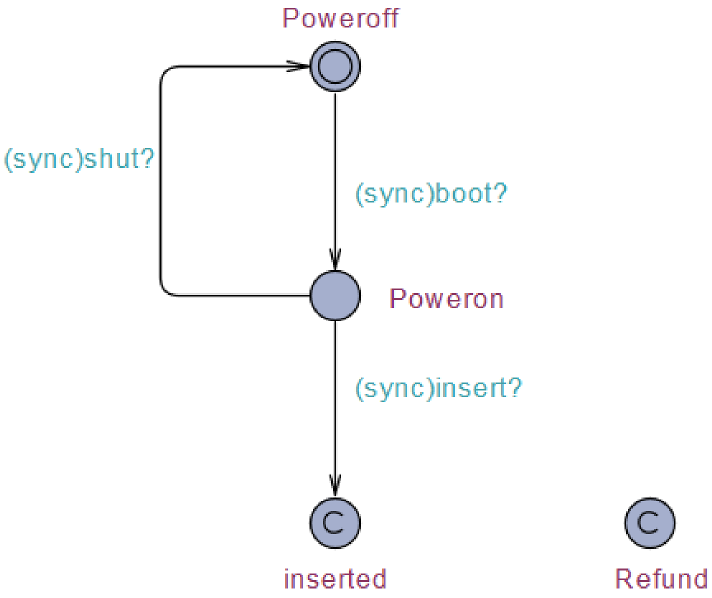
This module is designed to monitor and manipulate the status of coin acceptor with the following constraints:

1. Only one coin can be inserted at a time.
2. The module is capable of refund actions. It will refund on following conditions:
   (a) If the refund lever is pulled, one coin inside will be refunded. Nothing will happen if the acceptor holds no coin.
   (b) If the acceptor holds 100 coins, it directly refunds the coins inserted afterwards.
   (c) If the machine is not powered, it does not accept coins.
   (d) If the machine is turned off, the coin count resets without refunding.

Note:

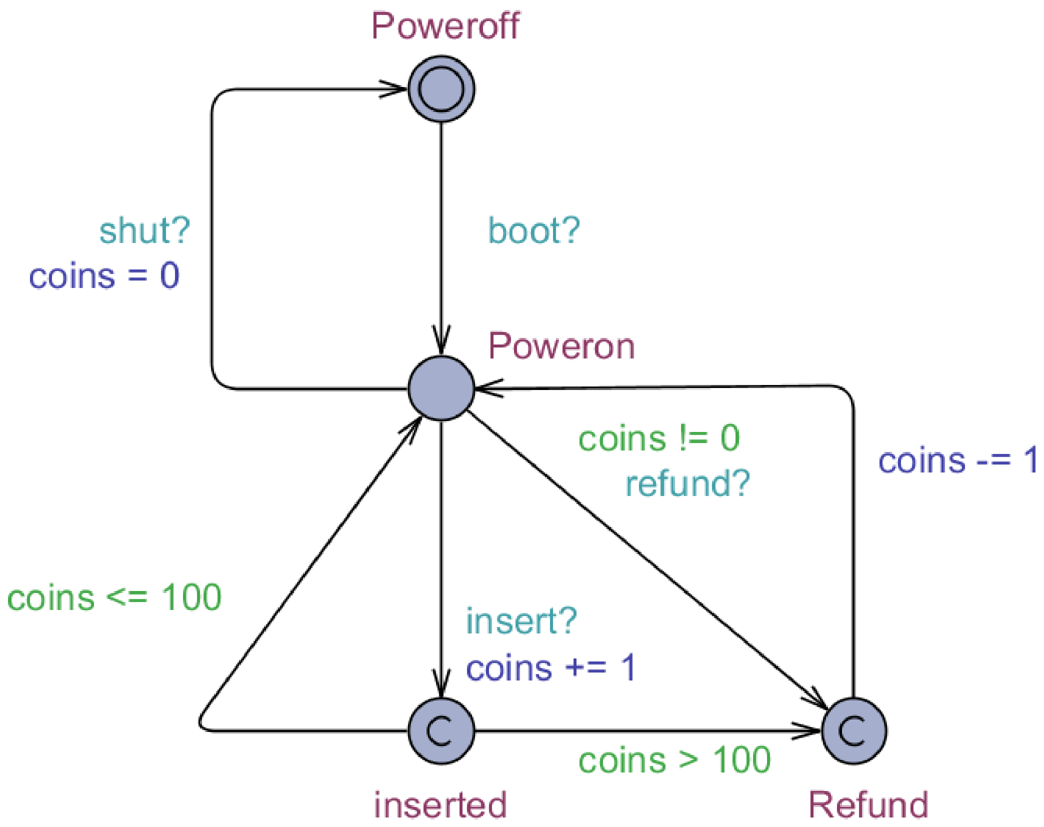
- When the machine is off, it physically rejects coins, but when the machine is on, you must take the committed state refund to return coins. An action of reducing coin count without passing refund in this situation will be considered wrong.





- You are allowed to add locations in this part of the question. Please make your graph neat if you decide to do so.
- You are allowed to add edges or add statements on existing edges.

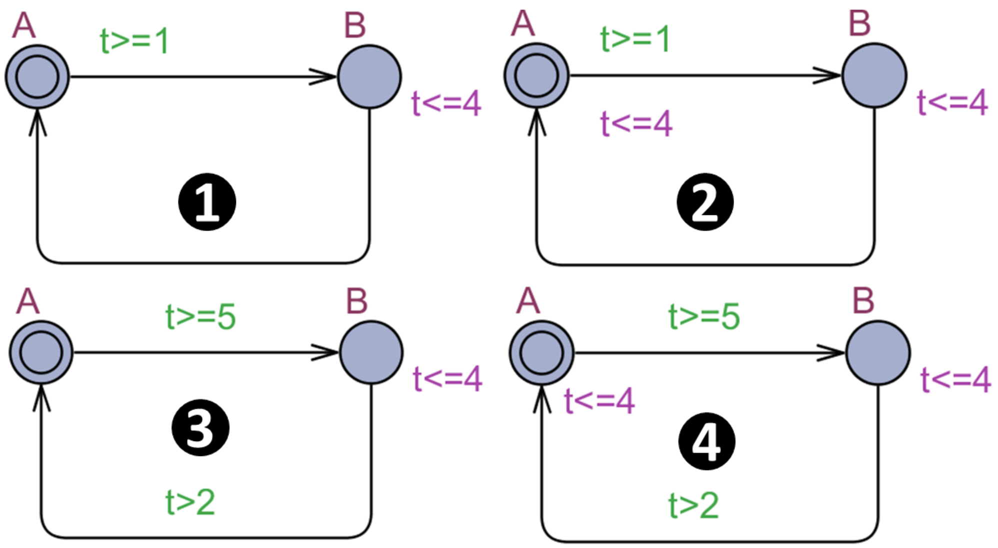
3) Write UPPAAL queries that can verify these requirements: (4pts)
1. The variable coins never exceeds 100, and never goes down to negative.
2. A game never lasts more than 10 minutes. (Assume that your first model is named host)



Problem 4. (8 pts)

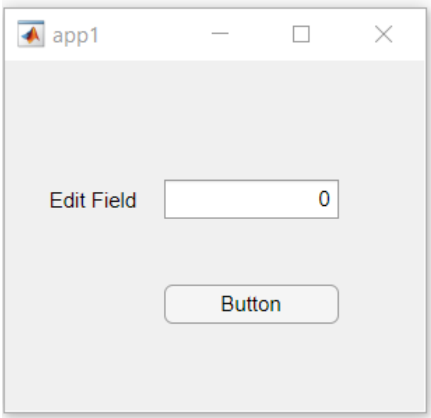
There are 4 UPPAAL models below with t as a clock:

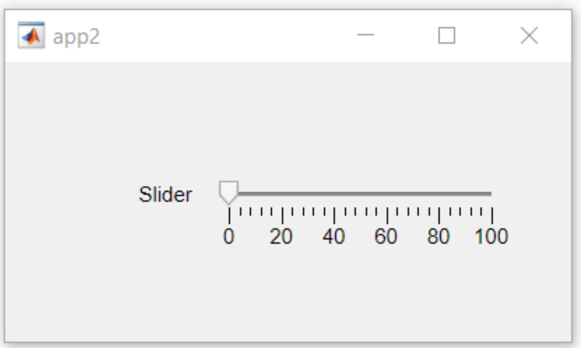
Please judge whether there exists deadlock in each model and explain why. **Answer:**
1) Yes. The model can stay in location A such that t is larger than 4, and location B cannot be entered. 2) No. 3) Yes. Location B cannot be entered when t is larger than 4. 4) Yes. Location A cannot be exited when t is equal to 4.



Problem 5. (12 pts)

We built two apps (app1 and app2) in Matlab as shown below:



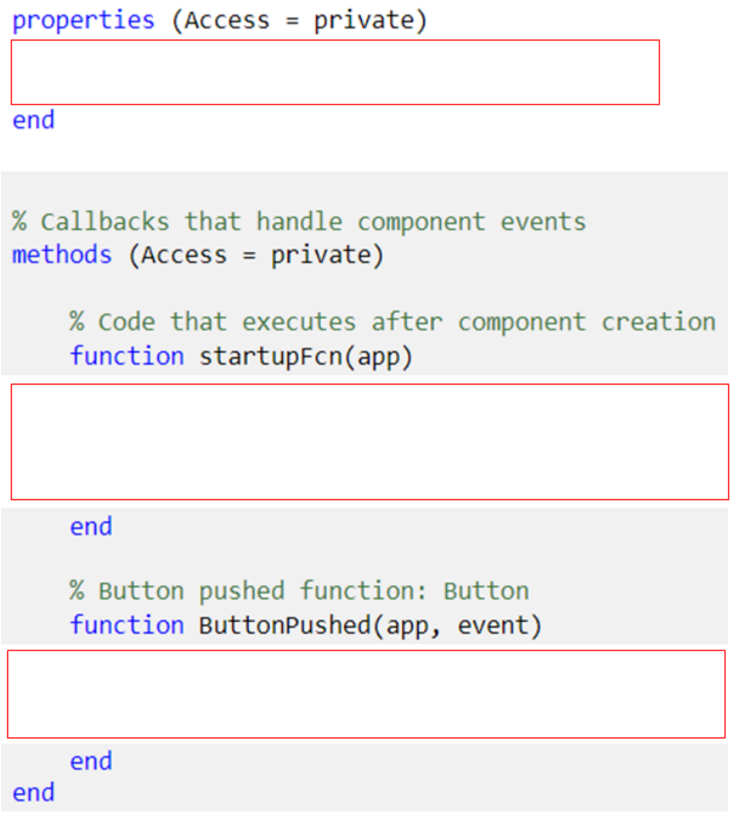


We would like to achieve the following functionalities:

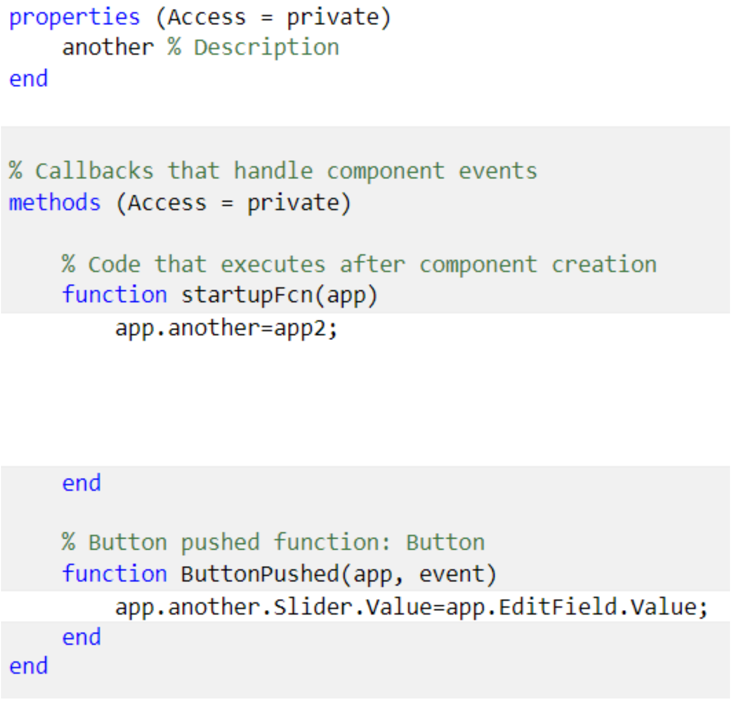
- Launching app1 (app1.mlapp) will launch app2 (app2.mlapp) automatically.
- Clicking the button on app1 (app.Button) will set the Value of slider on app2 (app.Slider) to the Value of edit field on app1 (app.EditField).

Please complete the code of app1 below so that the functionalities above can be achieved.

Answer:



```
properties (Access = private)

end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)

    end

    % Button pushed function: Button
    function ButtonPushed(app, event)

    end
end
```



```
properties (Access = private)
    another % Description
end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        app.another=app2;

    end

    % Button pushed function: Button
    function ButtonPushed(app, event)
        app.another.Slider.Value=app.EditField.Value;
    end
end
```