

CSS450: Assignment 1 Question Sheet

Instructions:

- In questions where you are asked to explain, please be concise.
- Show your work when necessary, be neat, precise, and brief!
- To help us grade your assignments and return this to you in a timely fashion please:
 - Put your name and answers in the answer sheet only (separate link provided). Anything you write outside of the answer sheet **will not** be graded.
 - Provide your answers in the order of the problems.
 - Please use only one side of 8.5x11 paper.
 - Please make sure you bring a hardcopy print out **of the answer sheet (!!only!!)** to submit at the beginning of class. Please do not print out the problems.

Your assignment may not be graded if any of the above is violated, you have been warned.

1. In the context of Model-View-Controller (MVC) architecture, GUI API is most suitable for implementing which of the component?

Controller!!: this is the component that the user interact with to change the Model. To support user interaction, we need “graphical user interface”

2. In our discussion of Left Mouse Button (LMB) click and dragging out a circle, which of the following(s) should **not** be a behavior defined for the circle?
 - a. `defineCircleCenter()`
 - b. `serviceLMBDrag()`**
 - c. `updateRadius()`
 - d. `handleColorChangeRequest()`
 - e. `supportLMBBehavior()`**
 - f. `increaseCircleVelocity()`

Remember to NOT define names of model according to user actions. BAD design.

3. Please refer to the provided Week-2 example 9: 2.9.MultipleModules:
 - a. Totally, how many ng-controllers are instantiated?

7 created: 1 parent, 4 for the sliders and 2 for the timers.

- b. What is the name of the enclosing parent controller?

exCtrl: yeah, bad name.

- c. How many instance(s) of the `mValue_1` variable is/are created?

1(!): multiple input devices are bind to this variable, but there is only one copy (it is the parent controller that defined this variable)

d. How many instance(s) of *css-timer mCallback* variable is/are created?

2(!): we know this because the left and right timer's services are from distinct functions!

4. I have a counting system where a count number can either be increased by pressing a button, or adjusted by a slider bar. I want to support 2-way data binding myself. Given the following code, I want to make sure that the echo output in *idMyEcho* and the slider position on *idSlider* are consistent when *idButton* is pressed and when the user changes the slider bar.

```
<input type="button" id="idButton" value="MyButton" onclick="incCount()">
<input type="text" id="idMyEcho" value="0">
<input type="range" id="idSlider" value="4" onclick="setCount()">
```

Here are the definitions for the *mCount*, *incCount()* and *setCount()*:

```
<script>
var mCount = 0;
incCount = function() {
  mCount++;
  var myEcho = document.getElementById("idMyEcho");
  myEcho.setAttribute("value", mCount);
  // → Part-A: What is the missing code?
```

Answer: update slider with the new count value

```
var aSlider = document.getElementById("idSlider"); // get the slider
aSlider.value = mCount; // set the value
```

NOTE: single line solution is perfectly fine:

```
document.getElementById("idSlider").value = mCount
```

```
};
```

```
setCount = function() {
  var aSlider = document.getElementById("idSlider");
  mCount = aSlider.value;
  // → Part-B: What is the missing code?
```

Answer: update echo with the new count value

```
var aEcho = document.getElementById("idMyEcho"); ← get the text output
aEcho.setAttribute("value", aSlider.value); // set the count value
```

// note the second line can also be:

```
aEcho.setAttribute("value", mCount);
```

NOTE: again, single line solution is perfectly fine, either of the following are correct:

```
document.getElementById("idMyEcho").setAttribute("value",
aSlider.value)
```

OR:

```
document.getElementById("idMyEcho").setAttribute("value", mCount)
```

```
};  
</script>
```

Please fill in the missing code for the above Part-A and B.

2pt for each correct answer.