ECE 413/513 - Fall 2019

Exam 3 Friday, December 06, 2019

Name: _____

No booklets, additional paper, textbooks, or other materials may be referenced during this examination with the exception of one 8.5"x11" sheet of paper (front and back). Your only other resources are a pen/pencil, eraser, and this exam.

Submit answers only in space indicated on each page.

1. Multiple Choice Questions (2.5 points each)

```
(1.1) A viewport is _____.
```

- (a) the width of the browser window
- (b) a percentage of the browser width or height
- (c) the maximum screen width that activates a media query
- (d) the visible area of a web page
- (e) None of the above
- (1.2) Given a device with a viewport size of 800x600 pixels and the following CSS, what will be the width of a <div> with class card-small?

```
.card-small {
   width: 20vw;
}
(a) 20 pixels
(b) 120 pixels
(c) 160 pixels
```

- (d) 600 pixels
- (e) None of the above

(1.3) Given the following Particle.publish(), which statement registers a response handler for the WebHook integration?

```
Particle.publish("thermoState", jsonPostData);
```

```
(a) Particle.handler("thermoState", respHandler, MY_DEVICES);
(b) Particle.handler("thermoState-response", respHandler, MY_DEVICES);
(c) Particle.subscribe("thermoState-response", respHandler, MY_DEVICES);
(d) Particle.subscribe("hook-response/thermoState", respHandler, MY_DEVICES);
(e) None of the above
```

(1.4) Which of the following statements creates a Timer that calls the function timerISR once every second?

```
(a) Timer smTimer(1, timerISR);
(b) Timer smTimer(1, timerISR());
(c) Timer smTimer(100, timerISR());
(d) Timer smTimer(1000, timerISR);
(e) None of the above
```

(1.5) Bootstrap is a _____ for creating responsive websites.

- (a) just a super fancy stylesheet
- (b) just a JavaScript library
- (c) back-end framework
- (d) front-end framework
- (e) None of the above

(1.6) Which of the following correctly sets the viewport width to the device width with an initial zoom of 100%?

```
(a) <meta name="viewport" width="device-width">
```

```
(b) <meta name="viewport" width="device-width" initial-scale="1">
```

(c) <viewport width="device-width" zoom="100%">

```
(d) <viewport width="device-width" initial-scale="1">
```

(e) None of the above

(1.7) In Bootstrap, the class col-md-4 means the _____.

- (a) column spans one third of the row's width when the browser's width is \geq 768px
- (b) column spans one third of the row's width when the browser's width is < 768px
- (c) row spans one third of the column's width when the browser's width is $\geq 768 px$
- (d) row spans one third of the column's width when the browser's width is < 768px
- (e) None of the above
- (1.8) Given the following HTML, where cookies-small.jpg is 300x200 pixels, cookies-medium.jpg is 600x400 pixels, and cookies-large.jpg is 1200x800 pixels, on a browser with a 3x screen, what is the image size in **logical pixels** when displayed in the browser.

- (a) 300
- (b) 400
- (c) 600
- (d) 1200
- (e) None of the above

2. Responsive Webpage (2.5 points)

Write a media query for a 600px breakpoint that make the following changes if the device width is less than 600px:

- Changes the width of the div with class container to 90% of viewport width
- Changes the display style for all elements with the class card to block

3. Bootstrap (7.5 points)

Using Bootstrap, given the HTML below, draw the rendered webpage assuming device screen widths of 600 and 1000 pixels. In Bootstrap, 600 pixels is considered a small device and 1000 pixels is considered a large device.

Note: Cards are just rendered with a thin rectangular border.

```
HTML
                                           Rendered Web Page (Box represents 1000-pixel screen width)
<div class="row">
 <div class="col-6 col-md-4">
   <div class="card">
     <div class="card-body">1.0</div>
   </div>
 </div>
 <div class="col-6 col-md-4">
   <div class="card">
     <div class="card-body">1.1</div>
   </div>
 </div>
 <div class="col-6 col-md-4">
   <div class="card">
     <div class="card-body">1.2</div>
   </div>
 </div>
</div>
<div class="row">
 <div class="col-8 col-md-6">
   <div class="card">
                                           Rendered Web Page (Box represents 600-pixel screen width)
     <div class="card-body">2.0</div>
   </div>
 </div>
 <div class="col-4 col-md-2">
   <div class="card">
     <div class="card-body">2.1</div>
   </div>
 </div>
 <div class="col-6 col-md-2">
   <div class="card">
     <div class="card-body">2.2</div>
   </div>
 </div>
 <div class="col-6 col-md-2">
   <div class="card">
     <div class="card-body">2.3</div>
   </div>
 </div>
</div>
```

4. State-based Software Design (15 points)

Given the following:

- An STL queue of sensor measurements, queue<double> sensorVals. As a reminder, an STL queue supports the following operations:
 - o .size() returns the number of elements in the queue
 - o .front() returns, but does not remove, the element at the front of the queue
 - \circ .pop() removes the element at the front of the queue
- Assume a function getResponseCode() returns the server response code for a Particle.publish() operation for a WebHook integration.getResponseCode() returns:
 - 0 if no server response was received yet
 - 200 if successful
 - \circ 400 if failure
- A WebHook integration exists for an event named dataPost that takes data in the following JSON format: o { sensorVal: 0.9874878 }

Draw a state diagram for a state machine that:

- 1. Executes every 250ms
- 2. If the sensorVals queue is empty, the state machine should wait until the queue is not empty.
- 3. When the sensorVals queue is not empty, the state machine should attempt to publish one measurement to the WebHook every second using the specified format.
- 4. Each time a measurement is published, the state machine should wait for up to 500 ms to receive a server response.
- 5. If the server response is not received or the response code is 400, the state machine should attempt to re-publish that measurement value, but only after 1 second has elapsed since the previous publish.
- 6. If the server response code is 200, the state machine should publish the next measurement, if another measurement exists in the queue, but only after 1 second has elapsed since the previous publish.