IS217 Final Exam

Answer the following questions in 1-2 paragraphs. Each one is worth 5 points.

1. What is a software design pattern? Why are they important?
2. A design pattern is like a blue print to anything. It is used in order to do some over and over for number of times. It is like a template to follow or a skeleton to begin with. They are like proven solutions to programmers.
3. What is unit testing? Why is it important? How would you use it?

Unit testing is a process in which the every bit of the function is tested in a program and check if it is giving the desired outcome. It is important as it is used to check the validations of the inputs, collaborations of different programs etc. I would use the unit testing in order to make sure that the information fed into the server is the way it should be. For eg, I should not have a number in my name column.

1. Describe the relationship between HTML, CSS, and JavaScript.
2. HTML, CSS and JS would combine to render us a DOM element. HTML gives the basic layout of the system. CSS takes care of the decorative stuff which is in the elements. JS will interact with the user and server in order to serve the best possible manner.
3. Describe the purpose of the Singleton design pattern.
4. Singleton Pattern restricts the instantiation of a class to just one object. The whole purpose of singleton pattern is to coordinate actions across the system with exactly one object.
5. Describe the purpose of the Factory design pattern.
6. Factory pattern is used to create objects. It is exactly like a factory. It creates object. It could create generic interface for creating objects.
7. Describe the purpose of the publish and subscribe pattern.

The purpose of Pub/Sub pattern is to notify the subscriber when the publisher has published and to notify the publisher that a user has subscribed the publisher.

1. Describe the purpose of the decorator pattern.
2. They are used to decorate the existing class. They are used to add behavior to the classes.
3. Write the JavaScript code that illustrates a decorator pattern.

// Decorator Pattern

// The constructor to decorate

function door() {

this.cost = function () { return 350; };

this.color = function () { return wodden-brown; };

}

// Decorator 1

function stopper(door ) {

var v = door.cost();

door.cost = function() {

return v + 25;

};

}

// Decorator 2

function knob( door ){

var v = door.cost();

door.cost = function(){

return v + 20;

};

}

// Decorator 3

function Insurance( door ){

var v = door.cost();

door.cost = function(){

return v + 20;

};

}

var mb = new Door();

stopper( mb );

knob( mb );

Insurance( mb );

// Outputs: door : 415

// Outputs: console.log( mb.cost() );

1. Write the JavaScript code that illustrates a factory pattern.

// FactoryExample.js

// Define a skeleton vehicle factory

function DoorFactory() {}

// Define the prototypes and utilities for this factory

// Our default FurnitureClass is door

DoorFactory.prototype.FurnitureClass = Door;

// Our Factory method for creating new Vehicle instances

DoorFactory.prototype.createDoor = function ( options ) {

if( options. FurnitureType === "Door" ){

this.FurnitueClass = Door;

}else{

this. FurnitueClass = addons;

}

return new this.FurnitueClass( options );

};

// Create an instance of our factory that makes cars

var door = new DoorFactory();

var car = DoorFactory.createDoor( {

DoorType: "Small",

color: "wooden",

addon: “none” } );

// Test to confirm our car was created using the vehicleClass/prototype Car

// Outputs: true

console.log( door instanceof Door );

1. Write JavaScript pseudo code that illustrates the singleton design pattern.

|  |
| --- |
| // Get the Singleton instance for creating one handle for one door if one exists |

|  |  |
| --- | --- |
|  | // or create one if it doesn't |

|  |  |
| --- | --- |
|  | getInstance: function () { |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | if ( !instance ) { |

|  |
| --- |
| instance = init(); |

|  |  |
| --- | --- |
|  | } |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | return instance; |

|  |  |
| --- | --- |
|  | } |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | }; |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  | })(); |

1. What is jQuery and provide examples of why you would use it? When would you not choose to you it?
2. jQuery is a JS library which is described as “write less, do more”. The purpose of jQuery is to simplify the JS on the website. It simplifies a lot of complicated things from JS such as AJAX, call DOM etc.
3. What is Backbone.js and how is it different than jQuery.
4. Backbone JS is a library of JS with RESTful JSON interface and is like the MVP (MULTIPLE VIEW PRESENTER) for single paged web apps. Jquery on the other hand is used in simplifying the code to be written in the script.
5. Write the JavaScript code to select an element by tag.
6. **var a = document.getElementByTagName(“\*tag name u r looking for\*”);**
7. Write the JavaScript code to select by ID
8. **var a = document.getElementById(“\*your\_element\*”);**
9. Write the JavaScript code to select an id and then add html to it.
10. **var a = document.getElementById(“\*your\_element\*”);**

**a.innerHTML = “<h1>Inner HTML example</h1>”;**

1. Write the JavaScript code to create an element.
2. **var heading = document.createElement(“h1”);**
3. What is Node.js?

Node.js is server side JS. It is the JS which communicates between the client and the server.

1. What is the difference between unit and functional testing?
2. Unit testing is testing all the functions in the program where as functional testing is when we test the program as a user.

Answer the following questions in 2-3 paragraphs. Each one is worth 10 points.

1. You have been hired to design and manage a team of developers tasked with creating a web application. How would you explain to your developer the importance of using standard design patterns when designing the system? Provide some practical examples that illustrate to your team how you will use the concept of design pattern within the project.
2. Given the scenario I would first of all give them a brief importance of team work. We have to work in a team in order to accomplish a task given to us. Thereafter, I would ask them if they have something in mind in order to see if they are familiar with the standard design patterns or they do their own work. After listening to all of them, I would insist on using the standard design pattern and give them few instructions.

As we are a team and the project is being assigned to a team, everyone has to be on the same page. Moreover, everyone has to perform a particular task in order complete their part of the work. The importance of design patterns is that when that work is being distributed to different team members in order to add or modify any codes, they know where to look and what to look. So patterns are very important to follow.

Lastly, I would make sure that all my team members are on the same page as I am. I would give them an example by giving them a piece of work individually and then combine it. It would be difficult for them and that would prove the point of why to use standard design-patterns.

1. You have been hired to design and manage a team of developers tasked with creating a web application. How would you explain to your developer the importance of creating unit tests? Provide some practical examples that illustrate to your team why unit testing is important.
2. Given the scenario, I would show them some bad examples of what happens if we don’t test the program properly. Unit testing is very important to the programmers as the whole world could see their mistake once it is online.

For one reason they don’t want to look stupid in front of the whole world regarding their credentials. I would also stress on functional testing. It is very important to check the system as if we were users. We can think in a different manner when we are on the other side of the development perspective.

I would exemplify by showing them work which was not proper. Like there are a few sites where one can enter a name instead of the zip code. We have to be smart enough to code the app in a manner that the user thinks that it is a smart app.

Bonus Points:

Create a repository on github and commit any file to it to demonstrate your ability to use Github. Include a link to the repository inside your test submission.