IS217 Final Exam

Khaled Elagamy

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?

They are reusable code solutions for common design problems. They are important because they let us use the previous experience from others and make our code better structured.

1. What is unit testing? Why is it important? How would you use it?

It’s a method to test unit source of code to ensure its fit to use, it works by isolating specific parts of code to show it is correct. I would use a test framework like Mocha.

1. Describe the relationship between HTML, CSS, and JavaScript.

HTML is the language to structure the content of a page, CSS, is to style the HTML elements visually, JS is for all the interactive functionality.

1. Describe the purpose of the Singleton design pattern.

It allows only one instance per class thus to a single object. It is useful when only one object is needed to coordinate others across a system.

1. Describe the purpose of the Factory design pattern.

It is a creational pattern that creates objects. And we can specify the type of objects we want to create.

1. Describe the purpose of the publish and subscribe pattern.

It uses an event channel which sits between the objects wishing to receive notifications (subscribers) and the object triggering the event (the publisher). It provides dynamic relationship between objects, and let an object be able to notify other objects.

1. Describe the purpose of the decorator pattern.

It dynamically adds alternate processing to objects.

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

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"**

**"http://www.w3.org/TR/html4/strict.dtd"**

**>**

<html lang="en">

<head>

    <title>*<!-- Insert your title here -->*</title>

</head>

<body>

    <script>

**function** Sale(price){

**this**.price **=** price **||** 100;

**this**.decorators\_list **=** [];

}

Sale.decorators **=** {};

Sale.decorators.fedtax **=** {

    getPrice**:** **function**(price){

**return** price **+** price **\*** 5**/**100;

    }

};

Sale.decorators.quebec **=** {

    getPrice**:** **function** (price){

**return** price **+** price **\*** 7.5**/**100;

    }

};

Sale.decorators.money **=** {

    getPrice**:** **function** (price){

**return** "$" **+** price.toFixed(2);

    }

};

Sale.prototype.decorate **=** **function** (decorator){

**this**.decorators\_list.push(decorator);

};

Sale.prototype.getPrice **=** **function**(){

**var** price **=** **this**.price,

    i,

    max **=** **this**.decorators\_list.length,

    name;

**for** (i**=**0; i**<**max; i**+=**1){

        name **=** **this**.decorators\_list[i];

        price **=** Sale.decorators[name].getPrice(price);

    }

**return** price;

};

**var** sale **=** **new** Sale(500);

sale.decorate("fedtax");

sale.decorate("quebec");

sale.decorate("money");

console.log(sale.getPrice());

    </script>

    Decorator Pattern

</body>

</html>

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

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"**

**"http://www.w3.org/TR/html4/strict.dtd"**

**>**

<html lang="en">

<head>

    <title>*<!-- Insert your title here -->*</title>

</head>

<body>

    <script>

**function** CarMaker(){

}

CarMaker.prototype.drive **=** **function**(){

**return** "Vroom, i have " **+** **this**.doors **+** "doors";

};

CarMaker.factory **=** **function**(type){

**var** constr**=**type;

**var** newcar;

**if** (**typeof** CarMaker[constr] **!==** "function"){

**throw**{

            name **:** "Error",

            message**:** constr **+** "doesn't exist"

        };

    }

**if** (**typeof** CarMaker[constr].prototype.drive **!==** "function"){

        CarMaker[constr].prototype **=** **new** CarMaker();

    }

    newcar **=** **new** CarMaker[constr]();

**return** newcar;

};

CarMaker.compact **=** **function**(){

**this**.doors **=** 4;

};

CarMaker.convertible **=** **function**(){

**this**.doors **=** 2;

};

CarMaker.suv **=** **function**(){

**this**.doors **=** 8;

};

**var** corolla **=** CarMaker.factory("compact");

**var** roadstar **=** CarMaker.factory("convertible");

**var** jeep **=** CarMaker.factory("suv");

console.log(corolla.drive());

console.log(roadstar.drive());

console.log(jeep.drive());

    </script>

    Factory Pattern

</body>

</html>

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

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"**

**"http://www.w3.org/TR/html4/strict.dtd"**

**>**

<html lang="en">

<head>

    <title>*<!-- Insert your title here -->*</title>

</head>

<body>

    <script>

**function** Universe(){

**var** instance **=** **this**;

**this**.start **=** 0;

**this**.bang **=** "bing";

            Universe **=** **function**(){

**return** instance;

            }

        }

**var** uni,uni2;

        uni **=** **new** Universe();

        uni2 **=** **new** Universe();

        console.log(uni **===** uni2);

    </script>

    singleton pattern

</body>

</html>

1. What is jQuery and provide examples of why you would use it? When would you not choose to you it?

It’s a JavaScript library that makes it easier to use JavaScript as it has a lot of prewritten code already. If I need more flexibility in my code, or do not want to link to the library.

1. What is Backbone.js and how is it different than jQuery

It is a library to structure and organize your code, it is different than jquery that jquery manipulates the DOM.

1. Write the JavaScript code to select an element by tag.

var mytag = document.getElementsByTagName("p");

1. Write the JavaScript code to select by ID

var myObj = document.getElementById("xyz");

1. Write the JavaScript code to select an id and then add html to it.

var myObj = document.getElementById("xyz");

myObj.innerHTML += “hello world” ;

1. Write the JavaScript code to create an element.

var btn=document.createElement("BUTTON");

1. What is Node.js?

It is JavaScript written and executed on the server side.

1. What is the difference between unit and functional testing?

Unit testing tests the smallest unit of code, typically a method/function

Functional tests usually check a particular feature for correctness.

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.

Patterns provide well-known solutions to reoccurring problems that developers are facing.

e.g: Say we have an e-commerce shopping Software, it has a component that handles sales transactions includes financial transaction: bank or a financial institution, stock changes and so on. In the system many other components need to know when a sale transaction takes place. If the sale component informs all the other components, Every time some new component needs sale information, we must update the sale component. The sale component will become dependent on many other totally unrelated components. The solution to this problem is Pub/Sub pattern The pattern where a consumer becomes a subscriber to specific information by listening to events.

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.

We are so focused on getting our code to work correctly, that we generally shy away from bad news, help with deadline pressures, all cause that we we will always end up with buggy junk that no one wants to admit ownership of. Instead unit testing offer you a "framework" -- a set of processes and code elements -- that will allow you to test your code more easily.

It Leads to a Better Design, let you understand how Your Code Works and gives you Confidence in Your Code.

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.