Ko Hing Chan

1. A software design pattern is a known solution to a programming problem. They can be creational, behavioral, or structural. They are usually written with instructions in the comments of the code in order for another programmer to understand their intent and apply it to their situation. Since some programming problems are recurring, instead of rewriting code from nothing, design patterns can be slightly modified to meet the needs of the problems.
2. Unit testing is when an individual module of code is run in order to confirm its functionality. Generally, it takes the smallest functional units and runs both independent and combined tests in order to ensure that there will be no breaks in the code. It is important to perform unit testing because it is a systematic method to find any breaks in the programming in order to fix it and in some cases optimize it. It would be best used to gain a different perspective on the code that one has written and I would use it to learn from my mistakes and learn how to improve my programming abilities.
3. HTML is hypertext markup language, which provides the canvas filled with text and divisions. CSS (cascading style sheet) is used to add color and schemes for how the text and divisions are laid on the canvas. Javascript is a scripting language that provides functionality and usability to the website and is very versatile with manipulating both HTML and CSS. It can provide the Backbone structure of the website as well as the server-side language where information for HTML and CSS can be stored.
4. A singleton pattern is a structural pattern that allows an object to pass through it and assigns it a unique class. The singleton pattern strictly assigns a class to only one object. This is useful in order to ensure that when a function calls a specific item that it does not operate on an array of other items that may share that same class. This is usually implemented as a necessary restriction in order to maintain order by reducing overlaps when necessary and maintaining constants for use across the board.
5. A factory design pattern is a creational pattern. The factory pattern allows the user to create objects that sometimes have certain basic properties based upon input. This pattern is best for creating multiple items of the same subset. For example, one could have a factory pattern to create several different cars with different makes and models. The main functionality of the factory pattern is that it can have all the objects and details stored without actually having to allocate memory to have them created, when not in use.
6. The Pub Sub pattern is a behavioral pattern. The publish-subscribe pattern allows users to subscribe to a channel of information that is managed and distributed by the publisher. The publisher send out information and those who are subscribed to that channel will receive it. This minimizes the amount of times the publisher has to output because the information will be sent out “as is” and the filter would generally fall on the user. For example, on forums, the subscriber is first subscribed to a base of popular channels and they can choose to unsubscribe or subscribe to other less popular, usually user generated, channels.
7. A decorator pattern is a structural pattern. The decorator pattern makes existing code more useful and promotes code reuse by reducing the need to sub-class because functionality can be added to the existing objects. It is called a decorator because it allows the user to modify the properties of objects without having to change a large portion of the code to do so. This is very useful for having to assign a specific property to an array of different objects in order to allow them to pass through a common function.
8. <script type = ‘text/javascript>

var itemDecorator = function(item) {

this.item = item;

};

itemDecorator.prototype = {

moves: function(){

this.item.moves();

},

talks: function(){

this.item.talks();

}

};

</script>

1. <script type= ‘text/javascript’>

function sandwich (stuff) {

this.meat = item.meat || "ham";

this.cheese = item.cheese || "american";

};

function drink (item) {

this.soda = item.soda || "coke";

this.water = item.water || "water";

};

function mealfactory() {};

mealfactory.prototype.lunchClass = sandwich;

mealfactory.prototype.createMeal = function(stuff) {

if (item.mealClass === "sandwich"){

item.mealClass = sandwich;

</script>

1. var something = function(){

var item1 = function(){

item2();

};

Var item2 = function(){

alert(‘but I’m hidden!’);

};

var frame = {};

frame.Singleton = function(){

item1();

item2();

};

frame.Singleton.prototype = {

if (item1() == item2());

create new class for item2();

};

};

1. jQuery is a javascript library made specifically for a simple, cross-browser compatible, client-side scripting language for HTML. jQuery would be best used for when creating an interactive user interface that will work on many browsers. I would use it when I need to create a website where I would like the same look and experience across the board such as an animated display of artwork where they could move through the objects on their terms.
2. Backbone.js is a slightly smaller library than jQuery designed for structure by providing a consistent framework. Much of backbone.js’s functionality is dependent on underscore.js a shorthand function library as well as require.js, which is also a library that helps improve shorthand notation of the MVC framework. The Model View Controller framework is the basis for all Backbone.js’s structure for single page web applications that contact RESTful servers. The model is the collection of all the objects. The view is what the users see. The controller is how the users can interact with the page. This is different from jQuery because of its smaller library it requires the user to follow a more intended pattern where jQuery is more open to more possibilities.
3. var M = document.getElementsByTagName(‘img’);
4. var N = document.getElementById(‘fire’);
5. var O = document.getElementById(‘fire’);

O.innerHTML(‘<p>oo</p>’);

1. var P = document.createElement(‘p’);
2. Node.js is the event-driven input/output server-side Javascript system. This also allows developers to create an entire web application with Javascript alone without having to rely on Apache or lighty. This software was written to enhance the usability and stretch the limits of the capabilities of Javascript. The node package manager has many program lib that allow the node applications to do a wide variety of processes such as graphical UIs, games, heavy computations such as in the site WolframAlpha.
3. Unit testing is the practice that evaluates the smallest portion of code that can stand alone. Functional testing is the evaluation of the cohesion of these units and how they work together to actually produce the desired results.
4. The concept of design patterns would be most important to know that they are out there for people to use. I would not want my team wasting their time creating their own code if they can easily modify a design pattern in order to create the desired result. They would have to understand that in the real world they will not have much time to waste and that they will have to manage their time and learn to learn another programmer’s instructions quickly. This will aid them in solving their coding problems in a timely manner with less hiccups.

For example, if they were to required to create many different sprites for a game in which they would have different traits and skills as well as many similarities in certain subclasses with base statistics. They would have to create or modify a factory pattern in order to create each race of characters. They would then be able to use a decorator to give each one of these characters special traits. They would then have to incorporate singletons in order to manage the race classes in order to allot bonuses that they might receive in game, accordingly. If they are not able to use design patterns effectively, they will not have the cohesion in the system of their code nor will they have the time to design the system themselves and they will be left in the dust and most likely fired for another person who can step up to the challenge.

1. Designing and developing a web application requires many parts to its creation. Many of these functional units may have to be assigned to different teams and brought together as one when the application is finished. To stress the importance of the need for unit testing is in that detail alone. I would first iterate to them that we are a team and even if we are working on different projects we have to work like one well oiled machine.

They would also have to understand that unit testing is the quality control of all the individual parts in this well oiled machine. If one part does not work, we will find it and fix it because it could potentially cause the machine (the team) to fall apart if it is not addressed properly. To ensure that we do not put together an app full of broken parts, we must conduct unit testing to ensure that each part meets our standards. Then and only then can the team pull together to create a truly amazing web application.