James Griffith

**Questions: 1 - 5 – 5 Points Each**

**1-2 paragraphs per answer**

1. What is a design pattern?

*A: A design Pattern is a reusable solution that is used to solve problems.  
 It is used to prevent minor mistakes that cause big problems in the software development*

*It is not an exact solution, but it is used as a solution template.*

1. How does the design pattern improve your ability to communicate with other programmers?

*A: The set of modern and classical design patterns were all agreed upon when created by programmers. The patterns are universal to all JS programmers and if something is wrong  
 with a program, a programmer can seek help and the other programmers will know what is going on because they are familiar with the common patterns.*

1. How does a design pattern help you solve programmatic challenges?

*A: Generally, the problems that a client may have already been solved before. Such as making a news feed or making a factory program. A programmer may see this and   
 think it would be good to give a bare bone template of the solution so others won’t have to start from scratch. The patterns definitely aid in development of programming.*

1. Is a design pattern a law or guideline? Why do you think so?

*A: A design pattern is a guideline because you don’t necessarily need to follow it. Many programmers build without even knowing what design  
 patterns are. I recommend that you follow the guidelines of a design pattern because it will make you design more simplistic and will aid in not repeating yourself or making minor mistakes.*

1. Do you think that you have improved as a programmer because of learning about them? How?

*A: I believe I have definitely improved as programmer by learning these because they help to understand what is needed in the real world and expose me to real, complicated problems.   
 I can be faced with a problem and immediately refer back to the design patterns for help, before this, all I knew was an IF statement.*

**Questions: 6 - 13 – 5 Points Each**

**1-2 paragraphs per answer**

**Explain the following design patterns in terms of their purpose and what type of pattern it is (Creational, Behavioral, or Structural)**

1. Factory Pattern

*A: Creational, makes instances of classes*

1. Observer Pattern

*A: Behavioral, observes the behaviors of the subscribers*

1. Decorator Pattern

*A: structural, assigns alternate methods to classes or objects.*

1. Singleton Pattern

*A: creational, uses a single class or object with global access points*

1. Constructor Pattern

*A: creational, creates objects*

1. Prototype Pattern

*A: Structural, Clones other objects*

1. Pub Sub Pattern

*A: Behavioral, with the proper event handler, you can subscribe and receive notifications from the publisher.*

1. Fly Weight Pattern

*A: Structural, it optimizes code that is repetitive. Speeds up the program*

**Questions: 14-17 – 5 Points Each**

**In JavaScript write an example of each pattern:**

1. Singleton Pattern

Person function {

Var human = “salman”

Alert(“hello, my name is “ + human);

If(human.length == 6)

Return human;

}

1. Factory pattern

tractorFactory(size, color) function {

This.color = “red”;

This.size= “big”;

}

1. Prototype Pattern

changeName.prototype cat function {

this.cat2 = “mary”;

Alert(cat2 + “ “ + “Says meow”);

}

1. Constructor Pattern

dog function {

This.dog1 =“sparky”;

Alert(dog1 + “ “ + “barks”);

}

1. Decorator Pattern

tractorFactory function {

This.color = “red”;

This.size=”big”;

}

tractorYard function(tractorFactory) {

This.color=”green”;

}

**Questions: 19-20 – 5 Points Each**

1. What is the difference between languages that uses classical inheritance vs. a language that uses prototypical inheritance? Can you name an example of a language for each respectively?

*A: A language that uses classical inheritance will inherit all that is in a class. An example of this would be java. Java may have a class called “Dog” and this will be the main class. Another class will be created called “dogNames” That will inherit the object dog in that class. It gives the program more optional.  
 A language that uses prototypical inheritance would be JavaScript. Prototypical inheritance clones all of the objects in a function and lets you use them in the prototype one.  
 You could have a function named “Cat” and then clone that function into a prototype pattern and have that function named “catNames”*

20.

1. Name two design patterns that work together well and provide an example of how you would use them in words or code.

*A: Two design patterns that would work well together would be the constructor and the prototype patterns.  
 The constructor pattern will create the objects, while the prototype pattern will clone or “inherit” the objects of the constructor pattern and give it other alternates of function.  
 cat1, in this case, was declared with the name of john at first. When the prototype pattern was introduced, John turned into mary. When the program is ran, an alert will show “john says meow” and then once it is exited, another alert will show with “mary says meow”*

Cat function {

This.cat1= “john”;

Alert(cat1 + “ “ + “says meow”);

}

changeName.prototype cat function {

this.cat1= “mary”;

Alert(cat1 + “ “ + “Says meow”);

}

**OUTPUT: john says meow**

**Mary says meow**

**In alert boxes**