Luis Mejia
CIS-14A-48590
Written Assignment 12

Please submit with both questions and answers in a single PDF file, word file, or ZIP archive.

1. **Using this object literal, create a constructor and instantiate this teacher as an instance of the constructor. var teacher = { firstname = “Ian”, lastname = “Lasky”, Degree = “MS”, Department = “BE”, Age = 47 };**

   Below is code for instantiating this teacher as an instance of a constructor using the object literal above. The object literal as a whole is passed to the constructor when instantiating.

```
// Object literal
var teacher = {
 firstName  = "Ian",
 lastName   = "Lasky",
 degree     = "MS",
 department = "BE",
 age        = 47
};
// Instructor constructor
function Instructor(params) {
 this.firstName  = params.firstName;
 this.lastName   = params.lastName;
 this.degree     = params.degree;
 this.department = params.department;
 this.age        = params.age;
}
// Instantiate this teacher
var ianLasky = new Instructor(teacher);
```

Luis Mejia
CIS-14A-48590
Written Assignment 12

2. **A constructor is a ____________.**

Constructors are functions that help us create objects much more easily and that adhere to the same design blueprint. Constructor functions are declared like normal functions with the difference they use the *this* keyword and their names begin with capital letters in order to distinguish them from normal functions. Constructors are defined once and then invoked as many times as needed to create that many instances of that constructor.

3. **What happens if you forget to use the keyword new with a constructor?**

If you forget to use the keyword *new* with a constructor when attempting to create an instance of an object, your code will produce an error. This happens because the *new* keyword is actually the one that creates the object in memory and without it, there will be no object for the constructor function to work with resulting in undefined.

4. **What does a constructor function return?**

A constructor function returns an object. The returned object is an instance of the constructor returning it. Objects are automatically returned from constructor functions so if you as the programmer return something else from the constructor function, you will get an error.

5. **What is an object literal? Show me an example of an object literal in Chapter 12.**

An object literal is a comma-separated list of name-value pairs inside of curly braces. They are also another way of creating objects. Below is an example of an object literal from Chapter 12.

```
// Object literal

var cadiParams = {

 make: "GM",

 model: "Cadillac",

 year: 1955,

 color: "tan",

 passengers: 5,

 convertible: false,

 mileage: 12892

};
```

Luis Mejia
CIS-14A-48590
Written Assignment 12

6. **What would you use if you wanted to know if an object was created by a specific constructor?**

   There are no object types in JavaScript and every object is of type object. If you wanted to know if an object was created by a specific constructor (or of a specific constructor), you would use the *instanceof* operator. The *instanceof* operator returns true if an object is in fact an instance of a constructor.

7. **List 3 builtin constructors that JavaScript comes with.**

   Javascript comes equipped with many builtin features. Three builtin constructors that JavaScript comes with are the Date, Array, RegExp.

8. **What keyword is used in a constructor to allow access to the object being constructed and adds properties to that object?**

   The keyword used in a constructor to allow access to the object being constructed and adds properties to that object is the *this* keyword. Without the *this* keyword, there would be no way for the *new* keyword to point to the new object being created.

9. **How are objects customized and initialized in Chapter 12?**

   In Chapter 12, objects are initialized using object literals and creating constructors and then invoking them. After an object is created, they can be customized. Using dot notation, new values can be given to existing properties or new properties can be added to the object. The *delete* keyword can be used to remove a property as well.

10. **When is it better to use a Constructor rather than an object literal?**

    It is better to use a constructor rather than an object literal when you want or need to create many different instances of the same constructor. Using constructors allows programmers to create objects much more easily that adhere to the same design blueprint. Constructors are better for larger scale while object literals are better for smaller scale.