

12/28/18

Lab 8 Scope and Encapsulation

CS472 Web Applications Programming

Implement the following in the IDE. You can send all the output to the console. Make some comment on the html page that the output is in the console.

Exercise 1:

Define a filter function on the String object. The function accepts an array of strings that specifies a list of banned words. The function returns the string after removing all the banned words.

Example:

```
console.log("This house is not nice!".filter('not'));
```

Output: "This house is nice!"

Exercise 2:

Write a BubbleSort algorithm on the Array object. Bubble sort is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted,

Example:[6,4,0, 3,-2,1].bubbleSort();

Output : [-2, 0, 1, 3, 4, 6]

Exercise 3:

Create an object called Teacher derived from a Person function constructor, and implement a method called teach that receives a string called subject, and prints out: [teacher's name] is now teaching [subject]. Create a Teacher object and call its teach method.

Also do the same thing using Object.create. When using Object.create you will need a factory function instead of a function constructor in order to pass parameters such as 'name' to be set in the prototype.

Exercise 4:

Write code that will create person, student, and professor objects.

- Person objects have
 - name and age fields
 - a greeting method that prints out: "Greetings, my name is [name] and I am [age] years old."
 - a salute method that prints out: "Good morning!, and in case I dont see you, good afternoon, good evening and good night!"

- Student objects inherit name, age, and salute from person. They also have a field 'major' and have their own greeting method. Their greeting is "Hey, my name is [name] and I am studying [major]. The greeting should be output to the console.
- Professor objects inherit name, age, and salute from person. They also have a field 'department' and have their own greeting method. Their salutation is "Good day, my name is [name] and I am in the [department] department." Output it to the console.
- Create a professor object and a student object. Call both the greeting and salutation methods on each.
- Do this exercise once using the object prototype approach for inheritance and then using the function constructor approach.