1. What is the difference between the following 2 statements?
   setTimeout(booyah, 2000);
   setTimeout(booyah(), 2000);

➔ Booyah with () immedialy calls the function rather than waiting 2000ms while booyah without () waits 2000ms to call the function

2. What do the following 2 alerts display (answer without running the code)?

```
var myfunc = function(a, x) {
 return a * x;
 };

var x = myfunc(2, 3);
 var y = myfunc;
 alert(x);
 alert(y(2,3));
```

➔ Both shows alert showing 6.

3. Write functions booyah1 and booyah2 so that in both cases below, an alert box comes up after 2 seconds that says “BOOYAH!”

```
setTimeout(booyah1, 2000);
setTimeout(booyah2(), 2000);

function booyah1(){
      alert("BOOYAH!");
}

function booyah2(){
return function () {
      alert("BOOYAH!");
   };
}
```

4. What is "Unobtrusive Javascript"? What is the practical application of Unobtrusive Javascript (and the reasons for using it)?

➔ Unobtrusive JavaScript is the way of writing JavaScript language in which we properly separate webpage content into 3 different concerns: content(HTML), presentation(CSS) and behavior(JS) aka MVC, thus allowing us to make a clear distinction between them.

For example:

HTML

In head tag,

<script src="file.js"></script>

In body tag

<button id=”btn” >Click Me</button>

JS (file.js)

```
window.onload = function() {

document.getElementById(“btn”).onclick = function() {

        alert(“Booyah”);

        };

};
```

Reason:

1. Separation of concerns i.e the HTML markup is now clean without any traces of javascript
2. Page load time is better
3. It is also easy to update the code as all the Javascript logic is present in a separate file.