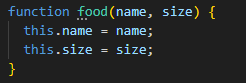
***JavaScript – Lab2 Report***

**Name:** Ahmed Mansour Mohamed  
**Track:** Open-Source Application Development

ـــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ

**Content: -**

1. Inheritance in Constructor Function
2. Abstract Class vs Interface
3. **Inheritance in Constructor:**

* In order to apply inheritance between two constructor functions, we are going to do it in 3 steps

**First**: create the parent constructor function, which other constructor functions will inherit from.

A screen shot of a computer code

Description automatically generated

A black screen with white text

Description automatically generated**Second**:we will write the child constructor function, though, writing this way, we can notice that name and size are redundant.  
So, we call the partent constructor function inside, using the .call() method, passing (this), which refers to the object wrapping the function, which is the child function, and give it also its arguments(name, size).

A black screen with white text

Description automatically generated**Third:** We would modify / add to the parent constructor function (food) some methods.

**But,** trying to use this method with the other child constructor functions, would give and error, as this method is added to the parent constructor function **prototype**, so, we would **Object.setPrototypeOf(parent\_proto, child\_proto),** and provide it with the parent prototype and child prototype to update it with the parent’s one.

**2-Abstract Class vs Interface:**

1. Abstract class can contain declaration and implementation of methods, while in interface we can only include the declaration part.
2. In JavaScript, we can’t achieve the concept of multiple inheritance through the classes, or abstract classes, but in case of interface, the class can inherit from a parent class and from an interface.
3. Abstract class can contain constructor, while interface doesn’t.
4. In abstract class, we can include many accesses specifier: public, protected, or private, but in interface we can’t, all inside the interface are public.
5. Interface performance is slower than abstract class performance.
6. Abstract class, like normal classes, can contain properties and methods, while interface can contain only methods.
7. In case of interface, we have to implement a al the methods that we declared in the interface.