function SportsCar() {
  this.refuel = function() {
    return "Refueling SportsCar with premium 94 octane gasoline";
  }
  this.mainTasks = function() {
    return "Spirited driving, looking good, driving to the beach";
  }
}

function CementTruck() {
  this.refuel = function() {
    return "Refueling CementTruck with diesel fuel";
  }
  this.mainTasks = function() {
    return "Arrive at construction site, extend boom, deliver cement";
  }
}

Note how the SportsCar and CementTruck objects do not define their own wheelCount and curbWeightInPounds properties and the associated accessor functions, as these will be inherited from the Vehicle object.

As before, you need a simple HTML page to test the new objects. Listing 5-6 lists the HTML page that will test these new objects. Pay special attention to the createInheritance function and how it's used to create the inheritance relationships between the Vehicle and SportsCar objects and the Vehicle and CementTruck objects. Also note that the describe function has been modified to attempt to access the wheelCount and curbWeightInPounds properties directly. Doing so should result in a value of undefined being returned.

Listing 5-6. classicalInheritance.html

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Classical Inheritance in JavaScript</title>
</head>
<script type="text/javascript" src="classicalInheritance.js"></script>

function createInheritance(parent, child) {
  var property;
  for(property in parent) {
    if(!child[property]) {
      child[property] = parent[property];
    }
  }
}
```