Class Inheritance in JavaScript Labs

Perform these labs on your own computer using Visual Studio 2022 or later, or VS Code 1.8x or later, to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Create an RV Class

Add a new folder named **classes** to your project.

Right mouse-click on the **classes** folder and add a new file named **RV.js**.

Define a class named **RV**.

Add the following public properties.

- rvid
- year
- make
- model
- listPrice

Add a **constructor** that accepts the same number of parameters as the number of public properties and assigns each parameter to the corresponding public property.

Create a file named **rvs**.**html** and add the following HTML into the new file.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width" />

  <title>RV Classes</title>

  <link href="styles/site.css" rel="stylesheet" />
</head>
```

```
<body>
  <header>
    <h1>RV Classes</h1>
  </header>

  <main>
    <p>Open your F12 Browser Tools to view the
results.</p>
  </main>

  // ADD A SCRIPT TAG HERE
  <script>
    'use strict';

    // Create an instance of an RV class
// Passing in 1, 2025, "Newmar", "New Aire", and 500000
to the constructor

    console.log("RV Object");
    console.log(rv);
    console.log("");
  </script>
</body>

</html>
```

Add a <script> tag to bring in the **RV.js** file.

Write the appropriate code to declare an instance of the **RV** class into a variable named *rv*.

Try It Out

Display the **rvs.html** file in your browser and your page should look like Figure 1.

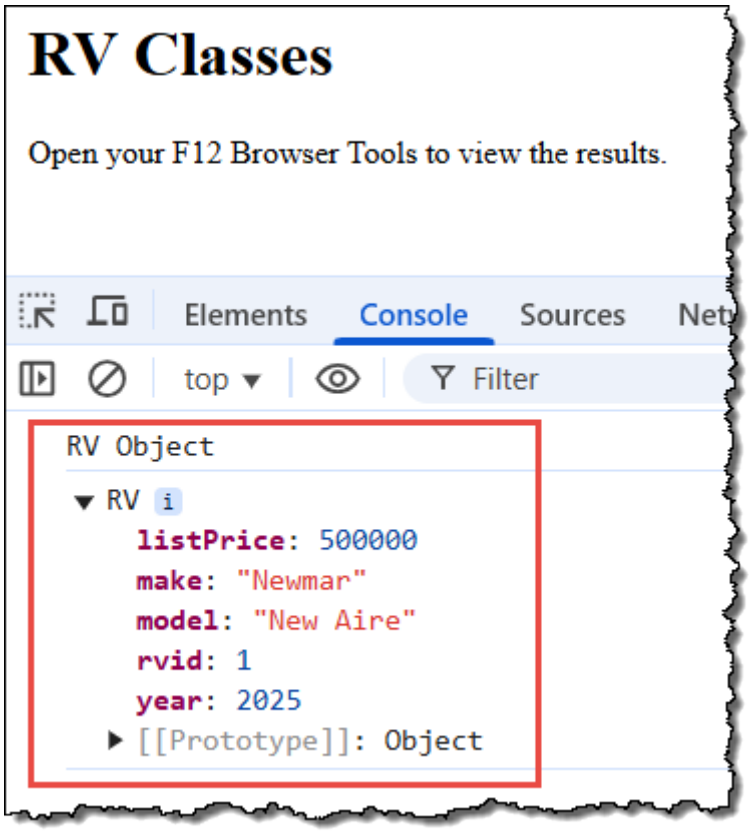

Figure 1: Create an RV class and assign some values to the public properties.

Lab 2: Create a Motor Home Class

Right mouse-click on the **classes** folder and add a new file named **MotorHome.js**.

Define a class named **MotorHome** that inherits from the **RV** class.

Add the following public properties.

- engineType
- class

Add a constructor that accepts the same parameters you defined for the RV class, plus *engineType* and *class*. Call the parent RV constructor passing in the appropriate arguments. Initialize the two new public properties in this class from the parameters passed to the MotorHome constructor.

Open the file **rvs.html** and add a <script> tag to bring in the **MotorHome.js** file.

Write the appropriate code to declare an instance of the **MotorHome** class into a variable named *mh*.

```
// Create an instance of a MotorHome class
// Passing in 1, 2025, "Newmar", "New Aire", 500000,
"Diesel", and "A" to the constructor

console.log("MotorHome Object");
console.log(mh);
console.log("");
```

Try It Out

Display the **findText.html** file in your browser and your page should look like Figure 2.

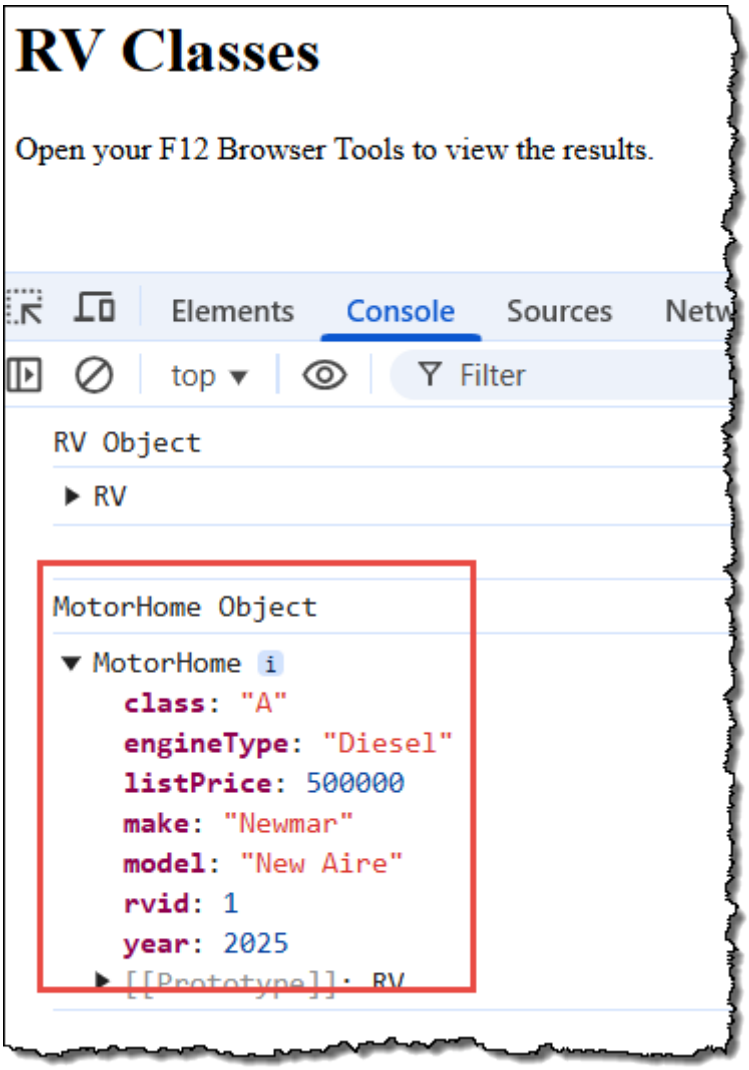

Figure 2: Create a MotorHome class that inherits from the RV class.

Lab 3: Create an RVTrailer Class

Right mouse-click on the **classes** folder and add a new file named **RVTrailer.js**.

Define a class named **RVTrailer** that inherits from the **RV** class.

Add one additional public property.

- hitchType

Add a constructor that accepts the same parameters you defined for the RV class, plus *hitchType*. Call the parent RV constructor passing in the appropriate arguments. Initialize the new public property in this class from the parameter passed to the RVTrailer constructor.

Open the file **rvs**.**html** and add a <script> tag to bring in the **RVTrailer.js** file.

Write the appropriate code to declare an instance of the **MotorHome** class into a variable named *mh*.

```
// Create an instance of a MotorHome class
// Passing in 1, 2025, "Newmar", "New Aire", 500000,
"Diesel", and "A" to the constructor

console.log("MotorHome Object");
console.log(mh);
console.log("");
```

Try It Out

Display the **xxx.html** file in your browser and your page should look like .

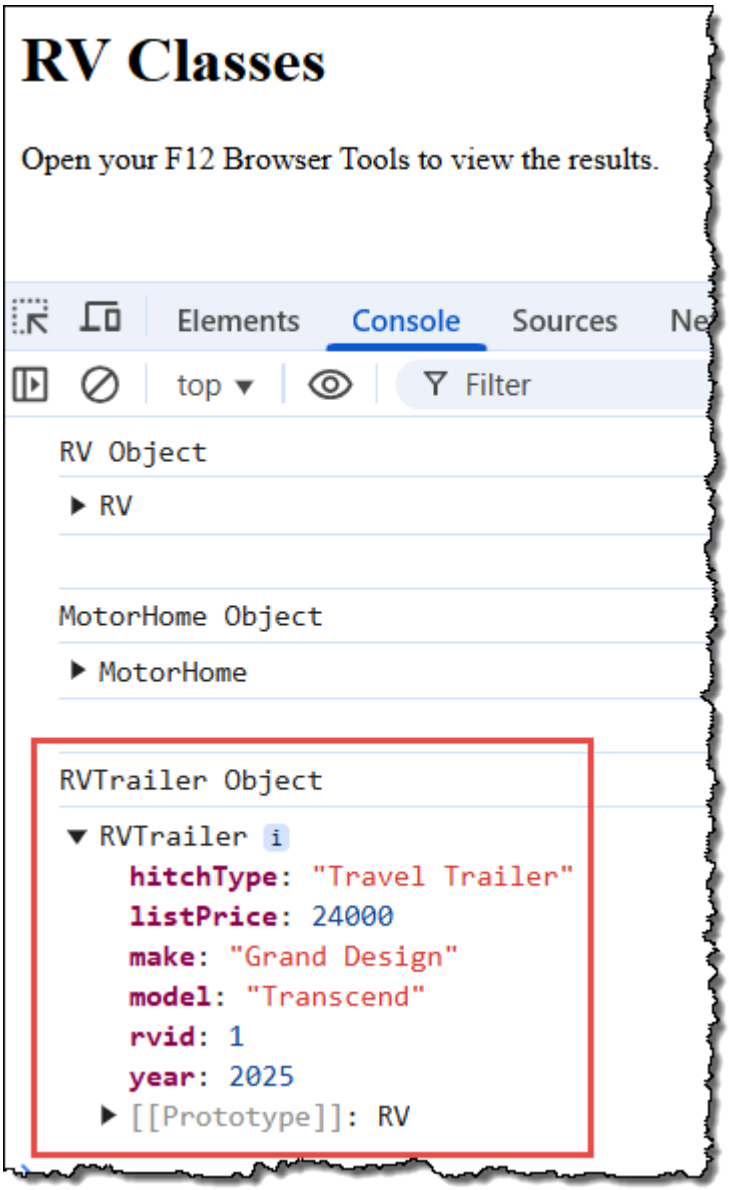

Figure 3: Create an RVTrailer class that inherits from the RV class.

Lab 4: Create a People Web Page (OPTIONAL)

Create a file named **people**.**html** and add the following HTML into the new file.

```
<!DOCTYPE html>
<html>
```

```
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width" />

  <title>People Classes</title>

  <link href="styles/site.css" rel="stylesheet" />
</head>

<body>
  <header>
    <h1>People Classes</h1>
  </header>

  <main>
    <p>Open your F12 Browser Tools to view the
results.</p>
  </main>

  <script src="classes/Person.js"></script>
  <script src="classes/Employee.js"></script>
  <script src="classes/Supervisor.js"></script>
  <script>
    'use strict';

    // Create an instance of a Person class
    let person = new Person("John", "Smith", 35);
    console.log("Person Object");
    console.log(person);
    console.log("");

    // Create an instance of an Employee class
    let emp = new Employee("Sally", "Jones", 29, 453,
60000);
    console.log("Employee Object");
    console.log(emp);
    console.log("");

    // Create an instance of a Supervisor class
    let sup = new Supervisor("Bruce", "Mulligan", 45,
42, 90000, [emp, new Employee("Ginger", "Rose", 25, 323,
55000)]);
    console.log("Supervisor Object");
    console.log(sup);
  </script>
</body>
```

```
</html>
```

The code within the <script> tag does NOT work. You are going to create three classes in the next lab to make this code display a page as shown in Figure 4.

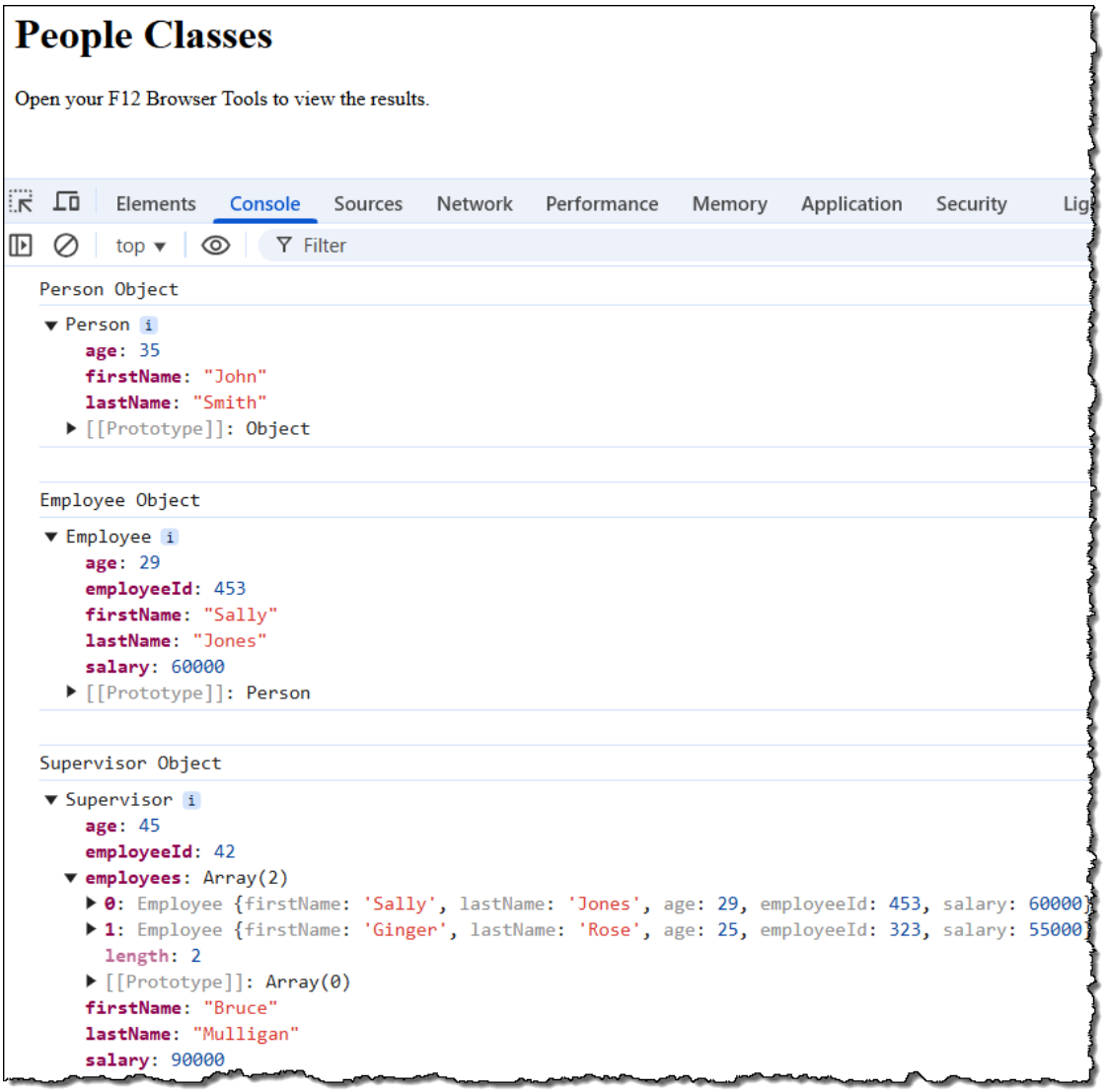

Figure 4: Create the appropriate classes to display this page.

Lab 5: Create the People Classes (OPTIONAL)

Create the Person Class

Right mouse-click on the **classes** folder and add a new file named **Person.js**.

Define a class named **Person**.

Add the following public properties.

- firstName
- lastName
- age

Add a constructor that accepts the same number of parameters as the number of public properties and assigns each parameter to the corresponding public property.

Create the Employee Class

Right mouse-click on the **classes** folder and add a new file named **Employee.js**.

Define a class named **Employee** that inherits from the **Person** class.

Add the following public properties.

- employeeId
- salary

Add a constructor that accepts the same parameters you defined for the Person class, plus *employeeId* and *salary*. Call the parent Person constructor passing in the appropriate arguments. Initialize the two new public properties in this class from the parameters passed to the Employee constructor.

Create the Supervisor Class

Right mouse-click on the **classes** folder and add a new file named **Supervisor.js**.

Define a class named **Supervisor** that inherits from the **Employee** class.

Add one new public property that is an array.

- employees

Initialize the *employees* property to an empty array.

Add a constructor that accepts the same parameters you defined for the Employee class, plus *employees*. Call the parent Employee constructor passing in the

appropriate arguments. Initialize the new public property in this class from the parameter passed to the Supervisor constructor.

Try It Out

Display the **people.html** file in your browser and your page should look like Figure 4.

THIS PAGE INTENTIONALLY LEFT BLANK

Answers

Lab 1

To the **RV.js** file add the following code.

```
// Define an RV class
class RV {
  // Define constructor
  constructor(rvid, year, make, model, listPrice) {
    // Initialize properties
    this.rvid = rvid;
    this.year = year;
    this.make = make;
    this.model = model;
    this.listPrice = listPrice;
  }

  // Public properties
  rvid;
  year;
  make;
  model;
  listPrice;
}
```

To the **rvs.html** file add a <script> tag

```
<script src="classes/RV.js"></script>
```

Add the following line of code to create an instance of the RV class.

```
let rv = new RV(1, 2025, "Newmar", "New Aire", 500000);
```

Lab 2

In the **MotorHome.js** file add the following code.

```
// Define a MotorHome class
class MotorHome extends RV {
  // Define constructor
```

```
  constructor(rvid, year, make, model, listPrice,
engineType, classType) {
    // Call parent's constructor
    super(rvid, year, make, model, listPrice);

    // Initialize properties
    this.engineType = engineType;
    this.class = classType;
  }

  // Public properties
  engineType;
  class;
}
```

To the **rvs.html** file add a <script> tag

```
<script src="classes/MotorHome.js"></script>
```

Add the following line of code to create an instance of the RV class.

```
let mh = new MotorHome(1, 2025, "Newmar", "New Aire",
500000, "Diesel", "A");
```

Lab 3

In the **RVTrailer.js** file add the following code.

```
// Define an RVTrailer class
class RVTrailer extends RV {
  // Define constructor
  constructor(rvid, year, make, model, listPrice,
hitchType) {
    // Call parent's constructor
    super(rvid, year, make, model, listPrice);

    // Initialize properties
    this.hitchType = hitchType;
  }

  // Public properties
  hitchType;
}
```

To the **rvs.html** file add a <script> tag

```
<script src="classes/RVTrailer.js"></script>
```

Add the following line of code to create an instance of the RV class.

```
let tt = new RVTrailer(1, 2025, "Grand Design",
"Transcend", 24000, "Travel Trailer");
```

Lab 5

Add the following code to the **Person.js** file.

```
// Define a Person class

class Person {
  // Define constructor
  constructor(firstName, lastName, age) {
    // Initialize properties
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
  }

  firstName;
  lastName;
  age;
}
```

Add the following code to the **Employee.js** file.

```
// Define an Employee class
class Employee extends Person {
  // Define constructor
  constructor(firstName, lastName, age, employeeId,
salary) {
    // Call parent's constructor
    super(firstName, lastName, age);

    // Initialize properties
    this.employeeId = employeeId;
    this.salary = salary;
  }
```

```
  employeeId;
  salary;
}
```

Add the following code to the **Supervisor.js** file.

```
// Define a Supervisor class
class Supervisor extends Employee {
  // Define constructor
  constructor(firstName, lastName, age, employeeId,
salary, employees) {
    // Call parent's constructor
    super(firstName, lastName, age, employeeId, salary);

    // Initialize properties
    this.employees = employees;
    this.salary = salary;
  }

  employees = [];
}
```