Using Classes

ECMAScript 6 introduces the concept of class available in traditional object-oriented languages. In ECMAScript 6, the class syntax is syntactical sugar on top of the existing prototype-based inheritance model. It does not add a new object-oriented inheritance model to JavaScript.

In this unit, you create an alternative implementation of the mortgage calculator application using a Mortgage class.

Part 1: Using a Class

1. Since this is an alternative implementation rather than the logical continuation of the previous implementation, make a copy of `index.html` and `main.js` in case you want to go back to that version.
2. In `main.js`, remove the `import` statement at the top of the file.
3. Add the following class definition at the top of file:
4. ```
   class Mortgage {

       constructor(principal, years, rate) {
           this.principal = principal;
           this.years = years;
           this.rate = rate;
       }

       get monthlyPayment() {
           let monthlyRate = this.rate / 100 / 12;
           return this.principal * monthlyRate / (1 - (Math.pow(1/(1 +
   monthlyRate),
                       this.years * 12)));
       }

       get amortization() {
           let monthlyPayment = this.monthlyPayment;
           let monthlyRate = this.rate / 100 / 12;
           let balance = this.principal;
           let amortization = [];
   ```
   ```

```
        for (let y=0; y<this.years; y++) {
            let interestY = 0;
            let principalY = 0;
            for (let m=0; m<12; m++) {
                let interestM = balance * monthlyRate;
                let principalM = monthlyPayment - interestM;
                interestY = interestY + interestM;
                principalY = principalY + principalM;
                balance = balance - principalM;
            }
            amortization.push({principalY, interestY, balance});
        }
        return amortization;
    }

}
```

5. Modify the **calcBtn** click event handler as follows:
6.
```
document.getElementById('calcBtn').addEventListener('click', () => {
    let principal = document.getElementById("principal").value;
    let years = document.getElementById("years").value;
    let rate = document.getElementById("rate").value;
    let mortgage = new Mortgage(principal, years, rate);
    document.getElementById("monthlyPayment").innerHTML =
mortgage.monthlyPayment.toFixed(2);
    document.getElementById("monthlyRate").innerHTML = (rate / 12).toFixed(2);
    let html = "";
    mortgage.amortization.forEach((year, index) => html += `
        <tr>
            <td>${index + 1}</td>
            <td class="currency">${Math.round(year.principalY)}</td>
            <td class="stretch">
                <div class="flex">
                    <div class="bar principal"

style="flex:${year.principalY};-webkit-flex:${year.principalY}">
                    </div>
                    <div class="bar interest"

style="flex:${year.interestY};-webkit-flex:${year.interestY}">
                    </div>
                </div>
            </td>
            <td class="currency left">${Math.round(year.interestY)}</td>
            <td class="currency">${Math.round(year.balance)}</td>
```

```
        </tr>
    `);
    document.getElementById("amortization").innerHTML = html;
});
```

7. On the command line, type the following command to rebuild the application:
8. `npm run webpack`

9. Open a browser, access http://localhost:8080, and click the **Calculate** button.

Part 2: Using Classes in Modules

To create the module:

1. Create a new file named `mortgage2.js` in the `js` directory.
2. Copy the `Mortgage` class definition from `main.js` into `mortgage2.js`.
3. Add the `export default` keywords in front of the class definition. `mortgage2.js` should now look like this:
4.
```
export default class Mortgage {

    constructor(principal, years, rate) {
        this.principal = principal;
        this.years = years;
        this.rate = rate;
    }

    get monthlyPayment() {
        let monthlyRate = this.rate / 100 / 12;
        return this.principal * monthlyRate / (1 - (Math.pow(1/(1 +
monthlyRate),
                this.years * 12)));
    }

    get amortization() {
        let monthlyPayment = this.monthlyPayment;
        let monthlyRate = this.rate / 100 / 12;
        let balance = this.principal;
        let amortization = [];
        for (let y=0; y<this.years; y++) {
            let interestY = 0;
            let principalY = 0;
            for (let m=0; m<12; m++) {
```

```
                let interestM = balance * monthlyRate;
                let principalM = monthlyPayment - interestM;
                interestY = interestY + interestM;
                principalY = principalY + principalM;
                balance = balance - principalM;
            }
            amortization.push({principalY, interestY, balance});
        }
        return amortization;
    }

}
```

To use the module:

1. In `main.js`, remove the Mortgage class definition.
2. Import the mortgage module. Add the following `import` statement as the first line in main.js:
3. `import Mortgage from './mortgage2';`

To build the project:

1. On the command line, type the following command to rebuild the application:
2. `npm run webpack`

3. Open a browser, access http://localhost:8080, and click the **Calculate** button.