CODE EXERCISE

CLASSES AND OBJECTS IN JAVASCRIPT

LAB 1

1) Start by understanding the concept of classes and objects. In JavaScript, a class is a template for creating objects (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods). Classes are a blueprint for an object, it describes what an object can do, but it doesn't create the object.

2) Create a Rock Band class using arrow function. The class should have the following properties: band name, band members and band genre.

```
let RockBand = class {
    constructor(name, members, genre) {
        this.name = name;
        this.members = members;
        this.genre = genre;
    }
}
```

3) Add a method to the class. A method is a function that is a property of an object. This method should display the band's information in the console.

```
let RockBand = class {
    constructor(name, members, genre) {
        this.name = name;
        this.members = members;
        this.genre = genre;
    }
    display() {
        console.log(`${this.name} is a ${this.genre} band, with members ${this.mem-
bers}`);
    }
}
```

4) Now that we have our class, let's create an instance of the class using the new keyword. Instantiating a class creates an object based on the class blueprint.

```
let metallica = new RockBand("Metallica", ["James Hetfield","Lars Ulrich","Kirk Ham-
mett","Robert Trujillo"], "Heavy Metal");
```

SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at http://bit.ly/CWDP2324.

2) Next, fill out the certification completion form at https://forms.gle/5EiUGCM6dGd1F2Py6.

Remember, that all of your exercises for each module should be included in a separate zip file.

GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at https://discord.gg/tgxX2fCrv5 and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at questions@dollardesignschool.com.

Remember to always to include which section your question is from and send any code you're working on!

CODE EXERCISE

CLASSES AND OBJECTS IN JAVASCRIPT

5) Now you can use the object and call the methods of the class

```
metallica.display();  // Metallica is a Heavy Metal band, with members James Hetfield, Lars Ulrich, Kirk Hammett, Robert Trujillo.
```

6) Repeat step 4 and 5 to create other rock bands

```
let pinkFloyd = new RockBand("Pink Floyd", ["David Gilmour","Roger Waters","Nick Mason","Richard Wright"], "Progressive Rock");
let ledZeppelin = new RockBand("Led Zeppelin", ["Robert Plant","Jimmy Page","John Paul Jones","John Bonham"], "Hard Rock");
pinkFloyd.display();
ledZeppelin.display();
```

7) Now you've learned how to create a class and object in JavaScript, and use arrow functions,You can also modify the class by adding or changing the properties and methods as needed.

LAB 2

JavaScript classes can be used to create subclasses, which inherit properties and methods from their parent class, and can also add new properties and methods of their own.

A subclass is created by defining a new class that extends an existing class using the extends keyword. The subclass inherits all the properties and methods of its parent class, and can also add new properties and methods of its own.

Here's an example of a parent class called Vehicle and a subclass called Car:

```
class Vehicle {
    constructor(make, model) {
        this.make = make;
        this.model = model;
    }
    startEngine() {
        console.log("Engine started");
    }
}
```

SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at http://bit.ly/CWDP2324.

2) Next, fill out the certification completion form at https://forms.gle/5EiUGCM6dGd1F2Py6.

Remember, that all of your exercises for each module should be included in a separate zip file.

GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at https://discord.gg/tgxX2fCrv5 and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at questions@dollardesignschool.com.

Remember to always to include which section your question is from and send any code you're working on!

CODE EXERCISE

CLASSES AND OBJECTS IN JAVASCRIPT

```
class Car extends Vehicle {
    constructor(make, model, color) {
        super(make, model);
        this.color = color;
    }
    honk() {
        console.log("Honk honk!");
    }
}
```

Create a simple web application that simulates a public transportation system. The application should have the following features:

1) A Vehicle class that represents a generic vehicle, with properties such as number, capacity, route, type (bus, train, subway, tram, etc.) and a method display that will show the type, number and capacity of the vehicle.

2) Subclasses for Bus, Train, Subway and Tram which should inherit from the Vehicle class and have additional properties specific to each type of vehicle.

3) A Passenger class that should have the following properties: name, age, destination, onVehicle (a boolean representing whether the passenger is on a vehicle or not) and a method display that will show the name, age and destination of the passenger.

4) Implement the following methods:

- addPassenger method, which allows adding a passenger to a vehicle.
- removePassenger method, which allows removing a passenger from a vehicle.
- listPassenger method, which should list the passenger and their destination.
- listVehicle method, which should list the vehicles and their properties.

5) Use of event listeners, to handle user input and any other user interactions with the app.

6) Use of error handling mechanism to validate the input, and show proper message when the user try to add a passenger that already on a vehicle, or remove a passenger that doesn't exist.

SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at http://bit.ly/CWDP2324.

2) Next, fill out the certification completion form at https://forms.gle/5EiUGCM6dGd1F2Py6.

Remember, that all of your exercises for each module should be included in a separate zip file.

GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at https://discord.gg/tgxX2fCrv5 and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at questions@dollardesignschool.com.

Remember to always to include which section your question is from and send any code you're working on!

CODE EXERCISE

CLASSES AND OBJECTS IN JAVASCRIPT

The technical requirements should include:

- Use of classes and objects to represent the various components of the transportation system.
- Use of inheritance to define subclasses of vehicles
- Use of constructors and methods to define and interact with objects
- Use of event

SUBMITTING EXERCISES

To earn certification all exercises must be submitted and accepted.

1) When you've completed all of the HTML exercises, please zip them into a single file and submit to our Dropbox at http://bit.ly/CWDP2324.

2) Next, fill out the certification completion form at https://forms.gle/5EiUGCM6dGd1F2Py6.

Remember, that all of your exercises for each module should be included in a separate zip file.

GETTING HELP

We always want to ensure that your questions are answered. There are a number of ways to get in touch.

1) We operate a lively Discord server. Join us at https://discord.gg/tgxX2fCrv5 and you can ask your question on Discord. Mark and our team of instructional assistants monitor this Discord and answer questions ASAP.

2) This certification program is offered on several platforms. Most platforms have a Q & A section where you can post questions. We monitor these and respond as quickly as we can.

3) You may use our question hotline email at questions@dollardesignschool.com.

Remember to always to include which section your question is from and send any code you're working on!