## FirstChair Software Developer Technical Challenge

Based upon your preference, **select one** of the following Challenges to complete. The first challenge focuses on Object Oriented Programming concepts. The second challenge focuses on JavaScript coding challenges.

If you get stuck, do the best you can to write comments and/or pseudo code in order to finish the problem. If you can't finish the program in its entirety, we still want to be able to see what programming concepts you do understand. Be prepared to show your code and run your program during your next interview.

### Challenge #1: Object Oriented Challenge

Using whichever object oriented programming language you prefer (preferably Java or Python), write a program that can solve the following requirements. You can write this program locally, or using a web-based Java Online Compiler such as Programiz.

**Requirements:**

1. Create a Class that represents a Zoo that you own. Here are some following characteristics of the Zoo Class:
    a. Holds the following information about the zoo:
        i. Name
        ii. City
        iii. Operating Hours
    b. Can keep track of one or more Animals that currently live in the Zoo. This includes:
        i. Ability to print out how many Animals are currently in the zoo
        ii. Ability to add an Animal to the Zoo
        iii. Ability to remove an Animal from the Zoo

iv. Ability to print out descriptions of all the Animals that currently live in the zoo

v. Ability to determine, depending on the current day of the week, the current Admission Price. Price is based on the following:
   1. Monday & Tuesday: $19.99
   2. Wednesday: $9.99
   3. Thursday & Friday: $19.99
   4. Saturday & Sunday: $25.99

2. Create another Class to represent an Animal that lives in the Zoo. Here are some following characteristics of the Animal Class:
   a. Holds the following information about the Animal:
      i. Animal Species
      ii. Name
      iii. Number of Legs
      iv. Gender
   b. Has the ability to print out a description of the Animal using the class level information
3. Create at 3 additional Classes that **extend** the Animal Class and represent 3 different species:
   a. Giraffe
      i. When a Giraffe is created, define the current height (in meters)
      ii. Has a method that can be used to compare how this Giraffe’s height compares to other Giraffes. For your reference:
         1. Male giraffes are an average of 5.5 meters tall
         2. Female giraffes are an average of 4.6 meters tall
   b. Crocodile
      i. When a Crocodile is created, define the number of teeth it has
      ii. Has a method that logs out its favorite food
   c. Giant Tortoise
      i. When a Giant Tortoise is created, define its current age

        ii. Has a method to determine whether the Giant Tortoise is young, middle-aged, or old. For your reference:
            1. Under 50 is young
            2. Between 50 - 100 is middle-aged
            3. Over 100 is old

4. Run the program to demonstrate that you have accomplished all the requirements.

**Challenge #2: JavaScript Challenge**

Using JavaScript, write a program that can solve the following requirements. You can use a web-based JavaScript Online Compiler such as Programiz to write and test your code.

1. **Array Cleaner:** You will be provided with an initial array (the first argument in the `arrayCleaner` function), followed by one or more arguments. Remove all elements from the initial array that are of the same value as these arguments. Note: You have to use the **`arguments`** object.

```
function arrayCleaner(arr) {

   // Add Logic
   return arr;

}

arrayCleaner([1, 2, 3, 1, 2, 3], 2, 3);
```

To test your code, if the following code snippet is run, you should get the following results:

```
console.log(arrayCleaner([1, 2, 3, 1, 2, 3], 2, 3); // Returns [1, 1]
console.log(arrayCleaner(["tree", "taco", 53], "tree", 53)); // Returns ["taco"]
console.log(arrayCleaner([2, 3, 2, 3], 2, 3)); // Returns []
```

2. **Spinal Case Converter:** Convert a string to spinal case. Spinal case is all-lowercase-words-joined-by-dashes.

```
function spinalCaseConverter(str) {
  return str;
}

spinalCaseConverter('CONVERT_This text To SpInal-Case');
```

To test your code, if the following code snippet is run, you should get the following results:

```
console.log(spinalCaseConverter("CONVERT_This text To SpInal-Case")); // Returns
“convert-this-text-to-spinal-case”

console.log(spinalCaseConverter("AllThe-small Things")); // Returns
“all-the-small-things”
```

3. **Capital Object Constructor:** Build out a constructor function for the Capital object that accepts a single string parameter that represents a Capital City and Province in which it is located (i.e. “Viedma, Rio Negro”). While the object should not have any properties, it should have the following 6 methods:
    a. getCity() // Returns City
    b. getProvince() // Returns Province
    c. getCityAndProvince() // Returns City and Province
    d. setCity(city) // Sets new City

e. setProvice(province) // Sets new Province

f. setCityAndProvince(cityAndProvince) // Sets new City and Province

If the following code snippet is run, you should get the following results:

```
const testCapital = new Capital('Viedma, Rio Negro');

console.log(Object.keys(testCapital).length); // Returns 6
console.log(testCapital instanceof Capital); // Returns true

console.log(testCapital.getCity()); // Returns “Viedma”
console.log(testCapital.getProvince()); // Returns “Rio Negro”
console.log(testCapital.getCityAndProvice()); // Returns “Viedma, Rio Negro”

console.log(testCapital.setCity(“Santa Rosa”));
console.log(testCapital.getCity()); // Returns “Santa Rosa”
console.log(testCapital.getCityAndProvince()); // Returns “Santa Rosa, Rio Negro”
console.log(testCapital.setProvince(“La Pampa”));
console.log(testCapital.getCityAndProvince()); // Returns “Santa Rosa, La Pampa”
```