**Lesson 2 Practice Hands-On**

**Directions**

For your Lesson 2 Practice Hands-On, you will be working with your knowledge on TypeScript classes and generics. This Hands-On will not be graded, but we encourage you to complete it. The best way to become a great programmer is to practice! Once you have submitted your project, you will be able to access the solution on the next page. Follow the below steps for setup.

**Setup**

1. Open up your terminal/command prompt.
2. Navigate to your desktop in your terminal.
3. cd Desktop
4. Next, navigate to the FullStackWeb directory in your terminal.
5. cd FullStackWeb
6. Then, navigate to the FEFAngular directory in your terminal.
7. cd FEFAngular
8. Create a new project folder named L02HandsOn located within the FEFAngular folder in your terminal.
9. mkdir L02HandsOn
10. Once the process is complete, navigate into the L02HandsOn directory:
11. cd L02HandsOn
12. Run the following to open your new project in VS Code (or you can open the folder within VS Code directly):
13. code .
14. Create a file named package.json and add the following code:
15. {
16. "name": "lesson-two-handson",
17. "version": "1.0.0",
18. "description": "lesson 2 hands on for the Angular course",
19. "main": "index.js",
20. "scripts": {
21. "start": "tsc && node index.js"
22. }
23. }
24. Create a file named index.ts
25. Within your command prompt/terminal (make sure you are within the L02HandsOn directory), run the following to create a tsconfig.json file:
26. tsc --init
27. Run npm start to create the index.js file.

**Requirements**

This project is broken into two parts. Please complete each part.

**Part 1**

1. To begin, add the following code into your index.ts file:
2. interface Person {
3. firstName: string;
4. lastName: string;
5. birthday: Date;
6. }
7. Create a new class named Entry.
   * This class should implement the Person interface above.
8. Add the following values into the Entry class:
   * a firstName value.
   * a lastName value.
   * a valid date for birthday.
9. Create a new instance of the Entry class with the name of newPerson. This new instance should include the following values as the correct types:
   * Albert, Einstein, 5/14/1879

Hint! You will need to use new Date()

1. Console.log the values of newPerson so the output looks like the following:
2. First Name: Albert, Last Name: Einstein, Birthday: Wed May 14 1879 00:00:00 GMT-0700 (MST)

**Part 2**

1. Create an array named numbers including the following elements:
   * 'twelve'
   * 'one-hundred'
   * 'sixteen'
   * 'forty-four'
   * 'seventy-three'
   * 'three-million'
2. Create a generic function named sortArray with the type of <S>.
   * This function should take in an array and sort the elements in alphabetical order.

Hint! You will need to use .sort() function

1. Add two console.logs:
   * One should console.log the array before it is sorted
   * One should console.log the array after it has been sorted

Hint! You will need to call the sortArray generic function

1. The output should be similar to the following:
2. Array before sorting: twelve,one-hundred,sixteen,forty-four,seventy-three,three-million
3. Array after sorting: forty-four,one-hundred,seventy-three,sixteen,three-million,twelve

**Example**

When finished, your project should look similar to the following:

