Solution Review: Finding Max in an Array

In this review, solution of the challenge 'Finding Max in an Array' from the previous lesson is provided.

We'll cover the following

Given solution

```
class FindMax {
    public static < T extends Comparable < T >> T array_max(T data[], int n) {
        T max = data[0];
        int i;
        for (i = 1; i < n; i++) {
            if (max.compareTo(data[i]) < 0) {
                max = data[i];
            }
        }
        return max;
    }
    public static void main( String args[] ) {
        Integer[] inputs1 = {2,8,20,3,4};
        Double[] inputs2 = {2.7,3.8,5.5,6.7,9.7};
        System.out.println(array_max(inputs1,5));
        System.out.println(array_max(inputs2,5));
    }
}
```

Explanation

- `public static <T extends Comparable<T>> T array_max(T data[], int n)`

  A static function is defined that extends the Comparable class (already implemented in Java) since we use a comparison function later in the program. `T` defines the generic user data-type. An array of generic type `T` is also defined as `T data[]`, where `n` is the size of `data[]`

- `T max = data[0];`

  Declares a variable `max` of generic type and store the element at 0th index in it.

- `for(int i = 1; i < n; i++)`

  for loop to traverse through the array `data[]`

- `if(max.compareTo(data[i]) < 0)`

  check if the `max` is smaller than the element at the ith index. If yes, then the built-in function `.compareTo()` returns -1 in this case.

- `max = data[i];`

  If the element at `ith` index is greater than `max` then replace the `max` with the element at the ith position.

- At the end return `max` to the calling point.

Let's wrap up this chapter by solving a quiz in the upcoming lesson.