Computer Science Virtual Learning

# HS Computer Science A

May 15th, 2020

Lesson: Free Response Friday

**Objective/Learning Target:**

Students will apply what they've learned this far in order to write code in response to an open ended free response question

# Free Response Question

A horse barn consists of N numbered spaces. Each space can hold at most one horse. The spaces are indexed starting from 0; the index of the last space is N - 1. No two horses in the barn have the same name. The declaration of the HorseBarn class is shown below.

Consider a software system that models a horse barn. Classes that represent horses implement the following interface.

```java
public interface Horse
{
    /** @return the horse's name */
    String getName();

    /** @return the horse's weight */
    int getWeight();
}
```

# Part A

Write the HorseBarn method findHorseSpace. This method returns the index of the space in which the horse with the specified name is located. If there is no horse with the specified name in the barn, the method returns -1.

For example, assume a `HorseBarn` object called `sweetHome` has horses in the following spaces.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| "Trigger" 1340 | null | "Silver" 1210 | "Lady" 1575 | null | "Patches" 1350 | "Duke" 1410 |

The following table shows the results of several calls to the `findHorseSpace` method.

| Method Call | Value Returned | Reason |
|---|---|---|
| `sweetHome.findHorseSpace("Trigger")` | 0 | A horse named Trigger is in space 0. |
| `sweetHome.findHorseSpace("Silver")` | 2 | A horse named Silver is in space 2. |
| `sweetHome.findHorseSpace("Coco")` | -1 | A horse named Coco is not in the barn. |

```java
public class HorseBarn
{
    /** The spaces in the barn. Each array element holds a reference to the horse
     * that is currently occupying the space. A null value indicates an empty space.
     */
    private Horse[] spaces;

    /** Returns the index of the space that contains the horse with the specified
     *  name.
     * Precondition: No two horses in the barn have the same name.
     * @param name the name of the horse to find
     * @return the index of the space containing the horse with the specified
     *       name;
     *       -1 if no horse with the specified name is in the barn.
     */
    public int findHorseSpace(String name)
    { /* to be implemented in part (a) */ }
}
```

# For More Resources, Solution Code and to Check Answers

Go to:  https://runestone.academy/runestone/books/published/apcsareview/ArrayBasics/horseBarnA.html