

CE Elementary ST CODE

| 10/24/2022

MISSING

10 Possible Points

Attempt 1



IN PROGRESS

Next Up: Submit Assignment



Add Comment

Unlimited Attempts Allowed

▼ Details

Searching

CE: Elementary ST CODE



Learning Objectives

- Create an elementary symbol table and fill it with multiple key-value pairs
- Identify symbol table methods from class BinarySearchST that are best suited to solve various challenges.
- Use symbol table methods to perform described tasks.



Overview

In this CE, you create a symbol table that associates pet IDs (whole numbers) with Pet objects. A series of challenges will ask you to identify methods from class BinarySearchST that are well suited for the specified task and to use them as you implement the

◀ [Previous](#)

(<https://slcc.instructure.com/courses/817632/modules/items/18753019>)

Submit Assignment

[Next](#) ▶

(<https://slcc.instructure.com/courses/817632/modules/items/18753022>)



Instruction

- Create a package called **cePet**.

Add two classes: **Pet** and **PetApp**, which includes the main method.

- Implement the class **Pet**, as described below.

It should be immutable and include the following:

- Three fields: name, age, and species
- A parameterized constructor that initializes all three fields
- Three getters (no setters)
- A toString method that returns a string of the following format:

{species} {name}({age})

Where {name}, {age}, and {species} are replaced with the actual field values.

E.g., dog Buddy(8)

- Implement the class **PetApp**.

- Create a symbol table of type BinarySearchST and call it **st**.

It uses pet IDs, which are whole numbers, as keys and associates with them Pet object.

Each pet has a unique ID.


When you are this far, [compare](#).

- Add the first entry to the symbol table.

The key should be a random number from the range [10, 30), and the pet is an 8-year-old dog named Buddy.

- Add nine more pets of your choice to the symbol table.

All keys should be a random number from the range [10, 30). However, make sure they are unique.

Find a method in [BinarySearchST](#)  (<https://algs4.cs.princeton.edu/31elementary/BinarySearchST.java.html>) that allows you to check whether a key is already in the symbol table. When you found one, [compare](#).

[← Previous](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753019>

Submit Assignment

[Next >](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753022>

- Print all the names of the pets in the symbol table in a single line.
When you are done, [compare](#).
- Check whether 10 is a key in the symbol table.
If so, print the corresponding pet.
If not, print a message that the given key was not found.

Repeat for the keys 17 and 23.

When you are this far, [compare](#).

- Find the smallest pet ID (no hard-coding)
Identify the method that is best suited for this task.
When you found it, [compare](#). Then implement the challenge.
- Find the smallest pet ID that is greater than 17 (no hard-coding)
Identify the method that is best suited for this task.
When you found it, [compare](#). Then implement the challenge.
- Find the number of pets that are in the symbol table.
Identify the method that is best suited for this task.
When you found it, [compare](#). Then implement the challenge.
- Add a 3-year-old dog named Waldi with pet ID 30 to the symbol table.
- Find the number of pet IDs that are smaller than 25.

Identify the method that is best suited for this task.
When you found it, [compare](#). Then implement the challenge.
- Print the pet with the pet ID 30.
- Find the second smallest pet ID and print the corresponding pet.
Identify the method that is best suited for this task.
When you found it, [compare](#). Then implement the challenge.

[◀ Previous](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753019>

Submit Assignment

[Next >](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753022>

Identify the method that is most helpful for this task.

When you found it, [compare](#). Then implement the challenge.

- Print all pets from the symbol table in straight columns.
The left column should list the pet ID (2 digits)
The right column should list the pet (name, age, species).
Include a header and proper spacing.



Sample Output

Notice that the pet IDs will differ from your output because they are randomly selected. The pets are likely to differ, too, because most of them were up to your choice. Make sure though that your output includes labels and full sentences, as demonstrated in this sample output. Also, use single empty lines to group related output.

```
All keys : 10 11 12 13 15 18 20 24 25 27
```

```
All pet names: Bobo Buddy Hazel Smokey Bella Leo Garfield Maddox Sylvester Wolf
```

```
Pet associated with 10: tortoise Bobo(50)
```

```
The key 17 was not found in the symbol table.
```

```
The key 23 was not found in the symbol table.
```

```
Smallest pet ID: 10
```

```
Smallest pet ID greater or equal to 17: 18
```

```
Number of pets: 10
```

```
Pet IDs smaller 25: 8
```

```
Pet with ID 30: dog Waldi(3)
```

```
Second smallest pet ID: 11
```

```
IDs  Pets
```

```
---  -----
```

```
10   tortoise Bobo(50)
```

```
11   dog Buddy(8)
```

[◀ Previous](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753019>

[Submit Assignment](#)[Next >](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753022>

- 24 dog Maddox(2)
- 25 cat Sylvester(7)
- 27 dog Wolf(9)
- 30 dog Strolch(3)



Submission

Create a screen recording following the [guidelines for lab recordings](https://slcc.instructure.com/courses/817632/pages/guidelines-for-ce-recordings) (<https://slcc.instructure.com/courses/817632/pages/guidelines-for-ce-recordings>).

The video should be **25-50 seconds** long.

Post the video.

 (<https://slcc.instructure.com/courses/817632/modules/items/18753011>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753012>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753013>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753014>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753016>)

 (https://slcc.instructure.com/api/v1/courses/817632/module_item_redirect/18753017)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753019>)

  (<https://slcc.instructure.com/courses/817632/modules/items/18753020>)

 (<https://slcc.instructure.com/courses/817632/modules/items/18753022>)

C

[← Previous](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753019>

Submit Assignment

[Next >](#)

<https://slcc.instructure.com/courses/817632/modules/items/18753022>

[◀ Previous](#)

[\(https://slcc.instructure.com/courses/817632/modules/items/18753019\)](https://slcc.instructure.com/courses/817632/modules/items/18753019)

Submit Assignment

[Next ▶](#)

[\(https://slcc.instructure.com/courses/817632/modules/items/18753022\)](https://slcc.instructure.com/courses/817632/modules/items/18753022)