AMACSS Study Group

Containers and Inheritance

Containers

What are Containers?

- Containers are objects
- We use them to store data, similar to lists and dictionaries
- Allow us to define the behavior of removals and insertions
- Last week we mentioned Stacks and Queues as ADTs
  - They are also containers

Container Methods

- Put/push – inserts an item into the container
- Get/pop removes an item from the container
- Is_empty – returns whether the container is empty or not

Other Common Methods

Peek – returns the item that would be returned by get

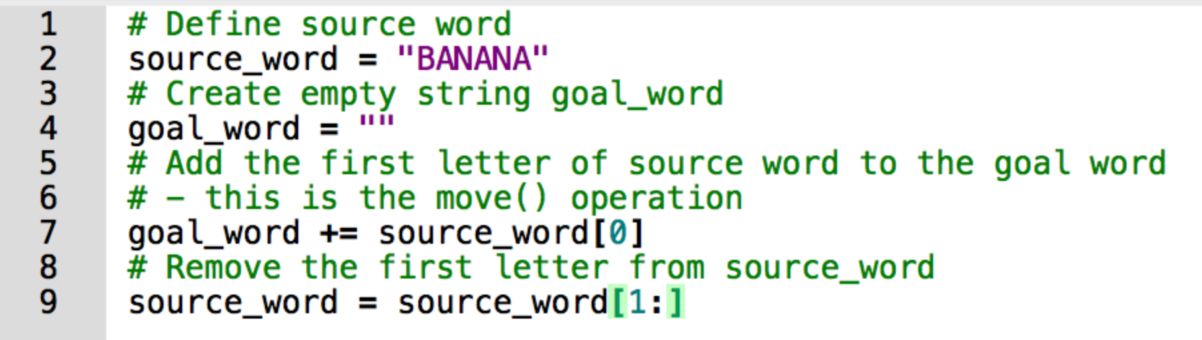
Copy – creates a copy of the container, with all the items in the container at the time of copying

Common Containers

- Queue
    - See week 1 notes
- Stack
    - See week 1 notes
- Bucket
    - Has a maximum capacity of 1
    - Same behavior as a queue/stack

Banana Game

- Take the source word, BANANA
  - Using ALL the letters from the source word, turn it into the goal word using a container
- Allowable moves: Put(), Move(), Get()
  - Move – takes the letter in source word at index 1 and adds it to the end of goal word
  - i.e



```
# Define source word
source_word = "BANANA"
# Create empty string goal_word
goal_word = ""
# Add the first letter of source word to the goal word
# - this is the move() operation
goal_word += source_word[0]
# Remove the first letter from source_word
source_word = source_word[1:]
```

Practice, Practice, Practice

Try the following examples using the specified containers

- “NICK” to “CNKI” using a stack
  - impossible
- “NICK” to “CNKI” using a queue
  - put(N) put(I) move() get() move() get()
- “PEACH” to “CHEAP” using a queue
  - impossible
- ”KNOG” to “KONG” using queue
  - move() put() move() get() move()

A Harder example

”CRABAPPLE” to “APPLECARB” using a Stueue

Answer:
Put(C) Put(R) Put(A) Get() Put(B) Put(A) Move()
Move() Move() Move() Get() Get() Get() Get()

Inheritance

What do you think Inheritance means?

- Parent-child relationship
- When a class inherits something, it can use the parents methods
- Eg:
  - I make a class, Person. This class has the method sayHello
  - I then make a class, Teacher. This class inherits Person, but I haven’t defined any methods for Teacher. By default Teacher has the method sayHello that it INHERITS from Person, even though we didn't define a sayHello method in Teacher

This doesn’t sound so clear, so lets implement it!

Lets practice inheritance

- For each of the following objects make a list of methods you would implement
  - Staff, Instructor, Student
- What sorts of relationships can we create? i.e can any object inherit from another?
  - Write down what object would inherit from another, and the methods it could inherit