

CS016, Spring 2009

Program 1: Stack

Out: Tuesday, 1/27

In: Tuesday, 2/3

Introduction

In this assignment, you will grow more comfortable with programming in Python and using IDLE, a Python IDE, by implementing a stack which increases and decreases its maximum capacity according to how much data it is currently storing.

Getting started

Detailed directions (including screencaps!) for installing this project and using IDLE are found in the “Setting up IDLE” handout, located at <http://www.cs.brown.edu/courses/cs016/Resource/stack.html>. You should install Stack by typing the command `cs016_install stack` into the shell.

This will install a new directory for you: `~/course/cs016/stack`. In this directory, you will find two files: `Stack.py` and `TestStack.py`. You will be editing `Stack.py`. You should not make any changes to `TestStack.py`.

Specifications

In lecture we have given you code for a fully-functioning stack which doubles its current capacity whenever that capacity is reached.

You should edit this code so that the stack will shrink to half its size whenever it can, i.e., whenever it is currently holding no more than half of its current capacity.

You will need to implement the following methods for your stack: `push`, `pop`, `isEmpty`, `peek`, `clear` and `print`.

Note: Unlike the other methods, your `print` method should be called `__str__` **not** `print`. Please see the lecture slides for an example.

You should also add documentation strings for each method you write. They

should follow the following rules:

- a) Say what the method does
- b) If it takes inputs, include “Inputs: <description of inputs here>”
- c) If it produces output, include “Output: <description of outputs here>”

IMPORTANT NOTICE

Python does not have a notion of private variables. However, it is a convention to prefix private variables with an underscore or two. Don't just copy the code directly from the lecture slides. You need to include underscores in private data names.

Testing your stack

Before you hand in your project, you will want to test it to make sure your stack is functioning properly. To do this, you can open the TestStack.py file in IDLE. From there, select “Run → Run Module”. If you return to the Python Shell, you will see how your stack performs under various tests. After each manipulation of the stack, the computer will output the following information:

Size: [how many items are currently in the stack]

Capacity: [how many items the stack can currently hold]

Head: [what the stack has stored as its head]

Data: [the items currently in the stack]

Underlying data: [the items currently in the stack plus the “empty” spaces in the stack, which are labeled as None]

Look through the output for these tests to see if your stack is meeting the specifications. You should be paying special attention to the size and capacity of your stack.

Handing in your finished product

Before you hand in this or any other project for CS016, you should make sure to create a README file. A README file is just a text file called “README” which you create using your favorite text editor and save to the directory of the project you are currently working on. You will not be able to hand in your project unless you create a README file.

To hand in your project, type `cs016_handin stack` into the shell.

IMPORTANT NOTICE

Python is a very convenient language. Its lists come with built-in stack methods. *Do not use these methods for this assignment!* You will not receive credit if you do.