CS 211
Lab 9 – Turtle's State Stack

1. Introduction

This lab assignment covers several Python programming concepts: classes, instance objects, and stack. The main exercise involves creating a State class representing (and saving) a turtle's state. The Stack will implement a FIFO (First-In, First-Out) data structure to save and retrieve turtle states. This mechanism is fundamental in the development of L-Systems capable of generating branching fractal structures.

1.1. The concepts covered in this Lab include the following:

- OOP in Python
- Stacks

1.2. Outcomes

- Implementation of the classes: `State` and `Stack`
- Understanding of the concept of FIFO data structure

1.3. Lab Outline

- Review of Python's classes and objects
- Review of Python's class variables
- Implementation of the following classes
  - `State`
  - `Stack`

2. Lab Activities

2.1. The Turtle State

In Project 9, Section 5.1, we justify why we need to save the turtle's state. Let us say the turtle executes the `commands` derived by the method `iterate` and finds itself at the point indicated by the red circle in Fig. 5 of Project 9's description. While executing the `commands` string, the `turtle` will stop there to draw the branch that extends to the right. It would be convenient to store the `turtle`'s state so it can come back to that state later. In drawing that branch, the `turtle` will end up who knows where, and it needs to come back to the state it was before.

The turtle state contains its position (`x`, `y`) and orientation (`heading`). Its constructor has the form

```
class State:
    def __init__(self, x=0, y=0, angle=0):
        pass
    def __str__(self):
        pass
    def __repr__(self):
        pass
    def set_state(self, t):
        pass
```

You are responsible for creating the `class state` and its methods (indicated above). Include this class in the file `state.py`.

2.2. Stack

Fig. 5 in Project 9's description shows a red circle where the turtle will start drawing the right branch. When the `turtle` follows that branch, it encounters the point indicated as "branch base", where it starts drawing a sub-branch. The `draw` method must save the `turtle`'s `State` at the red circle to restore it after drawing the branch. The `draw` method must save this second `State` at the branch base to restore it when it finishes drawing the sub-branch. Note that the second `State` must be restored before the first one. This scenario calls for a `Stack` data structure characterized by the property "Last-In, First-Out".

Implement a `Stack class` with the following methods:

```
class Stack:
    def __init__(self):
        pass
    def push(self, item):
        pass
    def pop(self):
        pass
    def is_empty(self):
        pass
```

The products you will develop in this lab are essential for Project 9. Include this class in the file `state.py`. Ensure you walk out of the lab having finished them and they pass the unit tests.

3. What to submit

Upload `state.py` to your Lab 9 workspace in Coding Rooms. Don't forget to upload it to the Project 9 workspace as well.