COMP 2002 – Assignment 1 (4%)

Due: 11:59pm, Tuesday, Oct. 3, 2023

Learning Objectives

The goal of this assignment is to gain experience with recursion and stacks.

Question 1: (30 pts)

Implement a recursive function, named `question1()`, that computes the mathematical function below.

$$f(m,n,p) = \begin{cases} m+p & \text{for } n < 1 \\ m+n & \text{for } p < 1 \\ f(m, \lg(n), \lg(p)) + f(m, n/2, f(m, n, \lg(p))) & \text{otherwise} \end{cases} \tag{1}$$

For example, input values $m = 5$, $n = 4$, $p = 3$ have an output value of $\simeq$144.546292.

Question 2: (70 pts)

Python classes have a number of methods with special names. We have seen one example of this already in this course: `__init__()` is the constructor for a class.

Implement the following special methods in the provided `ArrayStack` class. Follow proper programming etiquette – do not access private methods or variables of other objects.

`__eq__(self, other)`

Implements the == operation. Returns True if the current stack contains the same contents, in the same order, as the `other` stack, otherwise False. Does not modify this stack or the `other` stack.

`__ne__(self, other)`

Implements the != operation. Returns True if the contents of the two stacks are different or in a different order, otherwise False. Does not modify this stack or the `other` stack.

`__iadd__(self, other)`

Implements the += operation. Adds the contents of a second stack, `other`, onto the top of the current stack and returns the combined stack. Does not modify the `other` stack.

`__str__(self, other)`

Implements the str() operation. Returns a string representation of the contents of this stack. The string is formatted as [a, b, c, ...], where a, b, c, etc are the elements stored in the stack, with the top of the stack to the left. Does not modify the stack.

Example

```
>>> a = ArrayStack()
>>> a.push(5)
>>> a.push(10)
>>> a.push(15)
>>> b = ArrayStack()
>>> b.push(5)
>>> b.push(10)
>>> b.push(15)
>>> a == b
   True
>>> a != b
   False
>>> str(a)
   '[15, 10, 5]'
>>> c = ArrayStack()
>>> c.push(3)
>>> c.push(2)
>>> c.push(1)
>>> str(c)
   '[1, 2, 3]'
>>> a += c
>>> str(a)
   '[1, 2, 3, 15, 10 5]'
>>> str(c)
   '[1, 2, 3]'
>>> a == c
   False
```

Submission

Submit the code used to solve each question as a separate Python source file. Submit a single zip file containing all assignment submission files through the Assignment submission folder in Brightspace.

Python source code should be *.py plain text. The only file types allowed aside from Python source code (*.py) are pdfs and plain text (*.txt). Do not submit Word documents or rich text format documents. They will not be marked. Only submit a single zip archive. Do not submit files archived in rar format. That may result in your assignment not being graded.

Name all files with the format "firstname_lastname_studentid_...". Make sure to include your name and student ID as comments at the top of all Python source files.

Late submissions will be subject to a 10% penalty for each hour past the deadline.

Attribution

Submissions must represent your independent work.

If your submitted work includes unacknowledged collaboration, code materials, ideas or other elements that are not your original work, it may be considered plagiarism or some other form of cheating under MUN general regulations 6.12.4.2 and academic penalties will be applied accordingly.

The submission of work that has been created by generative artificial intelligence (GAI) tools and presented as a student's original work is considered an academic offence in this course. Using AI tools without proper citation constitutes plagiarism, and your work will be subject to the appropriate Memorial's Academic Misconduct policy.

If your submission contains any contribution by others, including internet sources and classmates, then you should include an attribution section detailing the extent of these contributions. This will also help distinguish what elements of the submission are original. You may not receive full credit if your original elements are insufficient, but can lessen penalties for plagiarism or copying if you acknowledge your sources.

Github

I encourage you to store and version your work on Github. It is good practice to do so as everyone uses git in the real world.

However, **it is a requirement that git repositories containing assignment material be private.** University regulations section 6.12.4.2 consider it cheating if you allow your work to be copied. There will be zero tolerance for this.