**GitHub Repository: https://github.com/dapwz/CS449**

**Sprint #0 Report**

Instructions

**Objectives**

* Make decisions on the SOS software development project.
* Learn unit testing and GUI programming in the language of your choice.

**Deliverables and Grading Policy**

Read the “CS 449 Homework Overview” document **carefully** and make the key decisions for the software development. Use the following template to complete your report.

1. **Key Decisions of the SOS Project (2 points)**

|  |  |
| --- | --- |
| Object-oriented programming language | Python |
| GUI library (strongly encouraged) | Tkinter |
| IDE (Integrated Development Environment) | VScode |
| xUnit framework (e.g., JUnit for Java) | unittest |
| Programming style guide (must read it carefully) | PEP8 |
| Project hosting site | Github.com |
| Other decisions if applicable |  |

Sample programming style guides:

* Google Java Style Guide: <https://google.github.io/styleguide/javaguide.html>
* Google C++ Style Guide: <https://google.github.io/styleguide/cppguide.html>
* Google Python Style Guide: <https://google.github.io/styleguide/pyguide.html>

1. **Unit testing (4 points)**

Find a tutorial on the unit test framework you have chosen and write at least two xUnit tests of a program you have written or found elsewhere. Attach here (1) the screenshot of your program execution and (2) the source code of your program.

A screenshot of a computer program

Description automatically generated

Basiccalc.py

class calc:

    def \_\_init\_\_(self, a, b):

        self.a = a

        self.b = b

    def getsum(self):

        return self.a + self.b

    def getdiff(self):

        return self.a - self.b

    def getmult(self):

        return self.a \* self.b

    def getlarge(self):

        if self.a < self.b:

            return self.b

        else:

            return self.a

basiccalctest.py

import unittest

from basiccalc import calc

class testcalc(unittest.TestCase):

    def testsum(self):

        calctest = calc(8, 2)

        self.assertEqual(calctest.getsum(), 10, 'The sum is wrong.')

    def testdiff(self):

        calctest = calc(8, 2)

        self.assertEqual(calctest.getdiff(), 6, 'The difference is wrong.')

    def testmult(self):

        calctest = calc(8, 2)

        self.assertEqual(calctest.getmult(), 16, 'The multiple is wrong.')

    def testlarge(self):

        calctest = calc(8, 2)

        self.assertEqual(calctest.getlarge(), 8, 'The larger is wrong.')

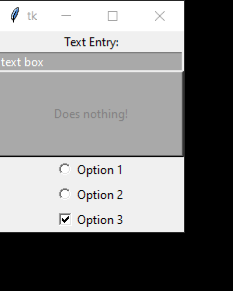
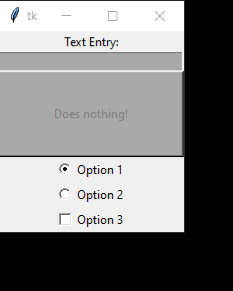
if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

1. **GUI programming (4 points)**

Write a GUI program in the language you have chosen for your SOS project. The GUI of your program must include text, lines, a check box, and radio buttons. While you are recommended to consider the GUI for the SOS game board, it is not required. In this assignment, any GUI program of your own work is acceptable.

Attach here (1) the screenshot of your program execution and (2) the source code of your program.



Basicgui.py

import tkinter as tk

window = tk.Tk()

# text

greeting = tk.Label(text="Text Entry:")

greeting.pack()

# textbox

entry = tk.Entry(fg="white", bg="dark grey", width=30)

entry.pack()

# button

button = tk.Button(

    text="Does nothing!",

    width=25,

    height=5,

    bg="dark grey",

    fg="grey",

)

button.pack()

# Radiobutton

selectedoption = tk.StringVar()

selectedoption.set('Value 1')

r1 = tk.Radiobutton(window, text='Option 1',

                    value='Value 1', variable=selectedoption)

r2 = tk.Radiobutton(window, text='Option 2',

                    value='Value 2', variable=selectedoption)

r1.pack()

r2.pack()

# checkbox

checkbox = tk.Checkbutton(window,

                          text='Option 3',

                          variable=selectedoption,

                          onvalue='Value 3')

checkbox.pack()

window.mainloop()