CISC106 Fall 2011 Lab10

- This lab an all subsequent labs will be due Sunday at 11:55 PM EDT on Sakai.
- Review the code examples from your notes in class.
- You may work with one or two other people on your lab (max size is three!). These people must be in your same lab section. If you do, **one** of you should be designated to submit the assignment on Sakai. **All of your names** should appear on code that you develop together[1].
- Whom do you think deducts more points: a happy TA, or a frustrated TA? Make your work easy to read! It isn't just good software engineering, it is good for your grade!
- EVERY python program/function must include header, doc string that contains a human-readable desciption of what the function does, and must be followed by a good series of tests, as discussed in class. Always test boundaries. Do not test erroneous input (e.g. a factorial function does not need to correctly handle strings).
- EVERY .py file must have a comment line at the very top containing your name(s), lab section, and a brief description of what the file is.
- Write the tests first! Real software engineers do this for very good reasons - so should you!

Programs (to be graded)

1. In this lab you will write a simple *Graphical User Interface* (GUI) application which takes some input from the user about a rectangle and then in turn displays some information about that rectangle back to the user. The first thing you should do is implement a `calc_area` function which takes in a $height$ and a $width$ and returns an area. This function should be unit tested (even though it's painfully simple!)
2. Next, import TKinter, tkSimpleDialog and tkMessageBox into your lab10.py. make a class called RectangleMainWindow which is a $Frame$, as such:

```
class RectangleMainWindow(Frame):
```

   The first thing your RectangleMainWindow's constructor should do is call Frame's constructor:

```
Frame.__init__(self, parent)
```

   This ensures you already have all the built-in Tkinter Frame functions available for you RectangleMainWindow. Next you'll want to call `pack` on the RectangleMainWindow to make sure it gets displayed properly to the screen:

[1] If you would like to work with someone but don't know whom, your TA may be able to help connect you to other students looking for lab partners.

```
        self.pack()
```

3. Now add these lines to lab10.py, beneath (and outside of) RectangleMainWindow:

```
if __name__ == '__main__':
    app = Tk()
    app.title('Rectangle info displayer')
    main_win = RectangleMainWindow(app)
    app.mainloop()
```

   These lines will set up a Tk application, gives it the title `'Rectangle info displayer'`, creates an instance of your RectangleMainWindow for the application and then runs the application's *event loop* (which is where control of the program is maintained until it's closed.) By this point, you should be able to run your application - see what it does. **Hint:** It doesn't do (or look like) too much.

4. Back to the RectangleMainWindow constructor. Add a Message to your window telling the user to enter a height and a width for their rectangle. Look back at `discalculator.py` from lab03 to see how to create a message and add it to your main window. Note how you specify a grid position with a row, column and *columnspan* (which tells the Message how many columns it should take up on the grid.)

5. Now add an Entry with a Label `'Height:'` in the row beneath the Message. Again, look at `discalculator.py` to see how to create the Label and then the Entry (and add both of these elements to the RectangleMainWindow's grid.) If you run your application, you should see the window now contains your message and the labeled input field. After this, you should add another field for wwidth below the one for height.

6. Now you should create buttons for displaying the rectangle's info, clearing the height and width fields, and quitting the application. Luckily, `discalculator.py` again has some good examples of buttons. Make sure all your buttons are in a row. Pay attention also to the `'command'` index of the button - this is what tell the program what function to run when the button is clicked.

7. Add a function `show_rectangle_info` to your RectangleMainWindow. This function is what will be called when the show info button is clicked. It should first pop up a dialog asking the user to input a name for the rectangle (look for `askstring` in `goldlib.py` from lab05 to see how to do this.) It should then show a pop-up displaying the rectangle's information, *e.g.*: `Rodney the rectangle is 100 meters squared in size.` if the user enters 10 for both the width and the height and `Rodney` for the name of their rectangle. `discalculator.py` or `goldlib.py` should have examples of how to do this - look for calls to `showinfo` in either of those. You can get the values of the height and width field with the *get* function (*e.g.* if you aptly named the height field `height` then you would call `self.height.get()`.) N.B. that both of the input fields should be attributes of the RectangleMainWindow. Also N.B. that get returns a string - you'll have to cast the field values as ints when handing them off to `calc_area`

8. Now all you need to do is take a few screenshots of your app in action and submit them along with your code.

You should submit your lab10.py, lab10_tests.py, a few screenshots of your GUI and any other docs required by your TA on Sakai.