

# CS 115, Autumn 2021

## Lecture 37: GUIs; lambdas; modules



Thanks to Marty Stepp and Stuart Reges for parts of these slides

# Exercise: drawing

- How can we draw a line that follows the mouse when the user holds it down and drags?
  - What mouse event types will we need to react to?

# Parameters and events

- What happens if we write the following?

```
hello['command'] = print_something("hi")
```

- This calls the function `print_something` passing it "hi" right now – it doesn't set it to be called when `hello` is clicked
- How can we pass parameters without parentheses?
  - We can't so we need lambda notation

# Lambda functions

- Lambda functions – small anonymous function that can be stored in variables
  - You can call another function from within one and have access to the variables in the function you declare it in

- Example:

```
from ECGUI import *

def print_something(name):
    print("hello " + name.get())

def main():

    window = make_window()
    hello = add_button(window, "hello")
    name = add_entry_box(window)
    hello['command'] = lambda: print_something(name)
    window.mainloop()
```

# Modules

- Module: a file containing definitions and statements
  - Usually functions and constants
  - Example: ECGUI.py
  - To use a module:
    - Put it in the same directory as your program
    - Include `from module_name import *` at the top of your program

# Importing

- So far we have used `from module_name import *` to import modules
  - This can be a problem if we import multiple modules that have functions with the same name
- Another option: `import module_name`
  - Imports the module but we must refer to all functions in it as **`module_name.function_name(parameters)`**
  - Prevents name conflicts but slightly longer less convenient syntax