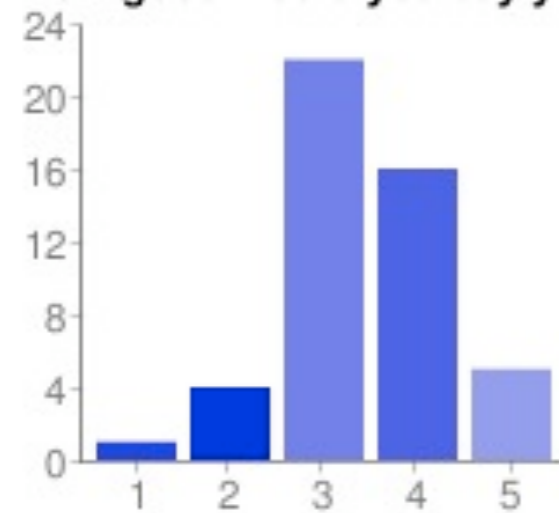


# Creative Computing

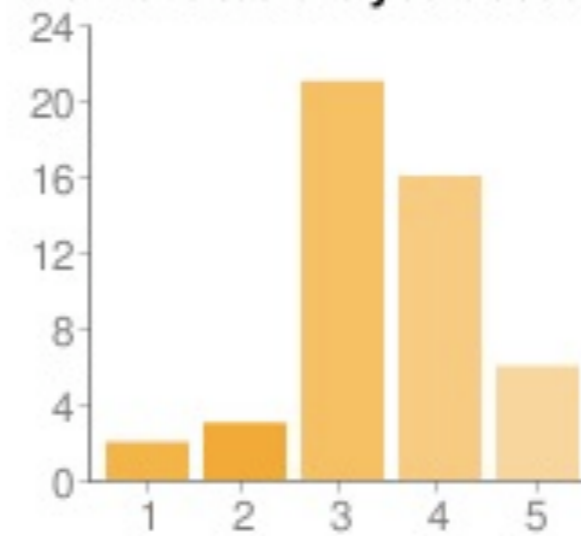
Surveys, variables, loops, math, print and style

# Surveys

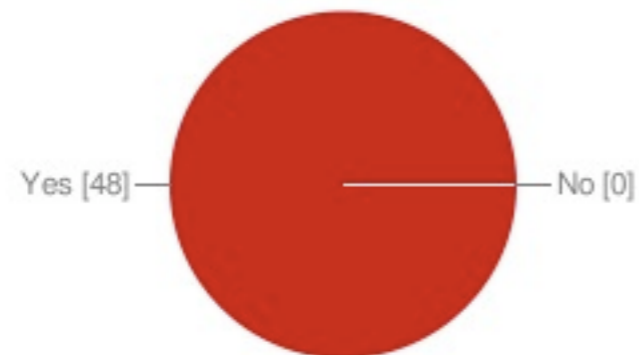
How good would you say you are at using computers



How excited are you about this class?



Do you have a computer at home?



# Surveys

- A lot of people are here accidentally but have great attitude
- Some night owls
- Lots of great hobbies: DJs, orienteering, guitar, dance
- People want to write: games, cellphone apps, apps for webcams, XO laptop

# Look forward to

- Learning card tricks on Friday
- Ms. Martin out of town October 1 and 2
  - You mess with the sub, I write you a referral no questions asked
- Microsoft guest speaker mid-October



# Tomorrow: Quiz

- Open notes
- You take it
- I tell you what you did wrong
- You get full credit
- Not a bad deal, right?
- A similar “real” one next week

# Grading Reminder

- Show up
- Have a good attitude
- Try hard and try lots of things, don't be afraid to make mistakes
- Ask for help
- You get an A or a B



# Style

- Group all functions at top
- Code that Python executes is at bottom
- Use as many functions as you can
- Aim for human readability
- Use comments on tricky bits

# Order of execution

- Python starts at the top
- Skips all definitions
- Executes commands sequentially, “jumping” into functions

# Parameters revisited

- When Python hits a function call with a parameter, it replaces parameter name with value

```
def forward(distance):  
    ...  
    ...  
  
def star(size):  
    for i in range(5):  
        forward(size) 10  
        right(144)  
  
forward(100)  
star(10)
```

The diagram illustrates the process of parameter replacement in Python. It shows two function definitions: `def forward(distance):` and `def star(size):`. Below these are two function calls: `forward(100)` and `star(10)`. Red arrows indicate the mapping: one arrow points from the value `100` in the `forward` call to the parameter `distance` in the `forward` definition; another arrow points from the value `10` in the `star` call to the parameter `size` in the `star` definition. A third arrow points from the value `10` in the `forward(size)` call inside the `star` function to the parameter `size` in the `forward` definition. The parameter names `size` and `size` are crossed out with red lines, and the values `100` and `10` are highlighted in red.

# Masking

- Be careful of using the same name multiple times!
- For example, can't use color as a parameter name if you want to also use the color function
- In this case, the color parameter takes over

# Python math

- + adding or string concatenation
- - subtraction
- / division
- \* multiplication or string multiplication

# Print

- A Python function
  - Display values to the user
  - Literal strings must be in quotes
  - Special characters
    - `\n` for newline
    - `\t` for tab
- ```
print(<value>)  
print(25 * 6)  
print("hello")
```

# Variables

- Store values in memory for later use
- Have a name
- Name must start with letter and can't have spaces (use \_ instead)
- Name should be descriptive

```
dollars_owed = 10  
tax = dollars_owed * .08
```

# for loops revisited

- `range(<number>)` is a function that gives back a list of numbers
- The `i` is a variable
- `i` gets a new value each time around the loop

```
for i in range(<number>):
```

```
...
```

```
...
```

# Your turn

- Code to calculate a cumulative sum?
  - Ex: `sum(5)` should print 15 because  $5 + 4 + 3 + 2 + 1 = 15$
- Give my stars a gradient
  - Use `color(0-1, 0-1, 0-1)`