1. Log in to your Sage Cloud account.

   (a) Start the Chrome browser.

   (b) Go to http://cloud.sagemath.com and sign in.

   (c) You should see an existing Project for our class. Click on that.

   (d) Click “New”, call it c05, then click “Sage Worksheet”.

In order to do sophisticated calculations, or to allow for multiple inputs, you will need to write programs. The classic “hello world!” program was our first example. It included a print statement. Other program features, in almost any language, include conditional statements (if..then..) and loops.

2. Type in the following function definition and evaluate.

   ```python
   def absolute(x):
       if x>=0:
           return x
       else:
           return -x
   ```

3. Now test it. Evaluate absolute(4), absolute(-4), etc.

4. Now use the program you just wrote in another program. Evaluate and test the following.

   ```python
   def abs_plus_five(x):
       return absolute(x)+5
   ```

5. You don’t have to add five, you can add any number by adding a parameter.

   ```python
   def abs_plus(x,y):
       return absolute(x)+y
   ```

6. Now test it. Try abs_plus(4,5), abs_plus(-4,5), abs_plus(-4,23), etc.
A string is a sequence of characters (letters, numerals, symbols, etc). If you put a sequence of characters between quotes, you are telling Sage to treat what’s between the quotes as a string (instead of as a keyword). Strings can be manipulated, and have places that can be filled in.

7. Type and evaluate `print 'This string has '{}''.format('17 characters')`. Now try replacing ‘17 characters’ with any other string.

8. Type and evaluate the following program.

```python
def superstring(x):
    print 'This string has {}'.format(x)
```

9. Now test your function. Type and evaluate `superstring('black letters')`.

More graphing and calculating basics.

10. Make a point at (4, 4) Evaluate `point((4,4))`.

11. Make it bigger by adjusting the “size” parameter. Get help with `point`?

12. Draw a line from (−1, 1) to (4, 4). Get help with `line`?

13. Make the line thicker by adjusting the “thickness” parameter.

14. Make the line dashed by adjusting the “linestyle” parameter.

15. Now make the line red.

16. Draw a triangle between (1, 1), (1, 2), and (2, 1) using the line command.

17. Now draw a triangle between (1, 1), (1, 2), and (2, 1) using the `polygon` command. What’s the difference?