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 c04, then click “Sage Worksheet”.

Programming basics.

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

2. Type in the following program and evaluate.

```python
def write_string(string_name):
    print string_name
```

Now type `write_string("hello world!")` and evaluate.

3. Type in the following function definition and evaluate.

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

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

5. Make a comment using “#” in the cell where you defined `absolute(x)` like “This function returns the absolute value”. Evaluate to check that Sage ignores it.

6. 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
```

7. 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
```
8. Now test it. Try \texttt{abs\_plus(4,5)}, \texttt{abs\_plus(-4,5)}, \texttt{abs\_plus(-4,23)}, etc.

A \textit{string} is a sequence of \textit{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 \textit{keyword}). Strings can be manipulated, and have places that can be filled in.

9. Type and evaluate \texttt{print ‘This string has ’ .format(‘17 characters’)}. Now try replacing ‘17 characters’ with any other string.

10. Type and evaluate the following program.

\begin{verbatim}
def superstring(x):
    print ‘This string has {}’ .format(x)
\end{verbatim}

11. Now test your function. Type and evaluate \texttt{superstring(‘black letters’)}.

\textbf{More graphing and calculating basics.}

12. Make a point at (4, 4) Evaluate \texttt{point((4,4))}.

13. Make it bigger by adjusting the “size” parameter. Get help with \texttt{point}?

14. Draw a line from (∀1, 1) to (4, 4). Get help with \texttt{line}?

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

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

17. Now make the line red.

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

19. Now draw a triangle between (1, 1), (1, 2), and (2, 1) using the \texttt{polygon} command. What’s the difference?