UW CSE 190p Section

6/21, Summer 2012

Dun-Yu Hsiao

Now it’s time to team up and code!

- Find a partner and make sure that you have at least one laptop.

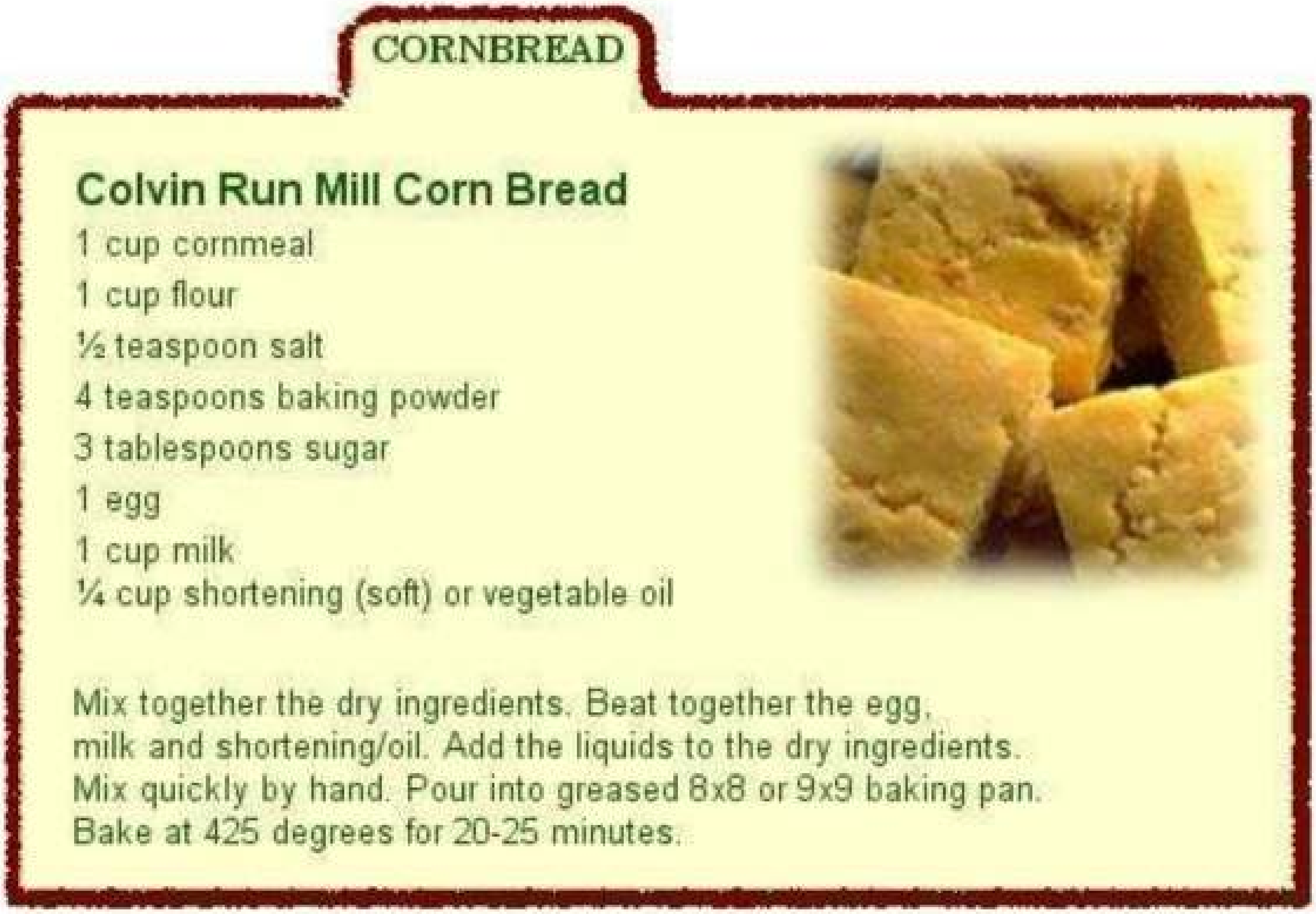
- Try to share what you think with your teammate!

- Today’s material will also be covered in the following lectures with more details, so don’t worry if you feel confused.

- When in doubt, you can always stop me and ask!

Continue from Michael’s lecture…

A program is a recipe



CORNBREAD

Colvin Run Mill Corn Bread

1 cup cornmeal

1 cup flour

½ teaspoon salt

4 teaspoons baking powder

3 tablespoons sugar

1 egg

1 cup milk

¼ cup shortening (soft) or vegetable oil

Mix together the dry ingredients. Beat together the egg, milk and shortening/oil. Add the liquids to the dry ingredients. Mix quickly by hand. Pour into greased 8x8 or 9x9 baking pan. Bake at 425 degrees for 20-25 minutes.

What is a program?

- A program is a sequence of instructions
- The computer executes one after the other, as if they had been typed to the interpreter

```
x = 1
y = 2

x + y

print x + y
print "The sum of", x, "and", y, "is", x+y
```

IDLE

- Now try to change x = 3, y = 4 and make print the result out again.
- Any better way rather than type everything all over again?
  - Use the editor
  - Be sure to save as .py to have code highlights

Exercise: Print a table

- Create a table using print about the simple info of your team.
- The required variable fields are: First name, Last name, Month of birth in number, and Favorite color.
- Your code should start with:

```
first_name = “Bill”
last_name = “Howe”
...
```

- Example output:

```
"Bill Howe, 1, likes green"
"Dun-Yu Hsiao, 5, likes red"
```

Exercise: Print a table

- Careful about the conversion between number and string

- Use str(some number)

Exercise: Convert temperatures

- Testing your program
- Making a temperature conversion chart
  - Chart the conversion of 5F, 32F, 104F, 212F, 293F
  - Print out example:

    ```
    5F -15.0C
    ...
    ```

  - (Tedious, isn’t it?)
- You can create a Python program!

Loops: basics

- Use loop to reduce code repetition!
- For loop:

```
for iterating_var in sequence:
    statements(s)
```

```
for x in [ 10, 2, 43]:
    print( x )
```

- List

```
list1 = ["a", "b", "c", "d"]
list2 = [1, 2, 3, 4, 5 ]
list3 = ['phys', 'chem', 1997, 2000]
```

Exercise: Convert temperatures

- Now try it using one for loop!

- Much more concise!

Exercise: Create a log table using loop

- Numbers:

  1, 2, 4, 8, 10, 20, 40, 80, 100, 200, 400, 800, 1000

- Import

  - To not reinvent the wheel!
  - Use the console to check usage quickly!

Careful!

- Don’t forget colon

- Careful about the indentation

Start Using Command Lines

- Command Prompt in Windows

- Terminal in Mac/Linux

Command Line Basics

- Show current directory/folder
  - pwd (on unix, linux, osx)
  - echo %cd% (on windows)
- List contents in the current directory/folder
  - ls (on unix, linux, osx)
  - dir (mostly only on windows)
- / on unix, linux, osx
- \ on windows

- Change directory
  - cd

  Use “tab” to loop through path!

- Make directory
  - mkdir (on unix, linux, osx)
  - md (on windows)

Exercise

- Go to your desktop directory
- In Desktop, create directories in this structure:
  - Desktop
    - test_dir1
      - test_sub_dir1-1
      - Test_sub_dir1-2
    - test_dir2
      - Test_sub_dir2-1

- Now go into test_sub_dir_1-2
  - Copy and save the commands you used in the report.
- Now go into test_sub_dir_2-1
  - Copy and save the commands you used in the report.

Invoking Python from the command line

- python myprogram.py
- python myprogram.py *argument1 argument2*

- The operating system command shell is *not* the same as the Python interpreter

Today’s takeaway

- IDLE
- Print
- Loop
- List
- Import

Questions?