



## Python Programming Task On Demand #2:

**TASK:** You have one period to create an interactive python program that demonstrates your ability of the basic programming skills we've worked on this month. **This will count as a test grade.**

Your program must meet the following criteria:

- Must have an original, coherent theme (cannot simply copy previous class exercises – but you can use your first TOD)
- Must include at least five comments to explain what each block of code does
  - o Include a title at the top and a short description that explains what the program is meant to do in one comment
- Must print at least five sentences to the terminal window
  - o At least three of those sentences must include variables that are printed
- Must collect at least three pieces of information from the user
  - o At least one piece of information must be a number that will be used in subsequent calculations
- Must perform at least one calculations
  - o At least one calculation must be done with a piece of information obtained from the user
- Must use at least **if & else** once each
- Must use at least one looping function (**for** or **while**)
- BONUS:** Must create and use at least one original function

This is an “open-notes” assessment. Feel free to use any programming materials you’ve created from this unit! Also, you get two “question” passes. You can use these to ask me a question without penalty.

**Submitting Your Task:** Once you have written your script (program), test it. If it works, email the script (the TextWrangler file) to Mr.Z at michael.zitolo@gmail.com. It is due at the end of class.



## SUGGESTIONS FOR THEMES

Your program must have a theme. If you've already thought of one, great! If not, here are some ideas:

1. Build off of your code from the first Programming TOD.
2. Create a program that McDonald's or another fast-food joint could use to take orders from customers. Ask the customer if they want certain items (like 2 or 3 food/beverage choices) and how many of those items they want. Then, determine how much their bill comes to, including tax.
3. Create a program that a teacher could use for attendance and grades. For example, the program could calculate how many students are present when the teacher enters how many students are absent. The program could also ask the teacher for some grades of a student and calculate that student's average.
4. Create a program that a college student could use to calculate what their student loan payments. You could ask the student how much money they want to borrow and over how many years they think they'll need to pay it back. The program could calculate how much interest the loan will accumulate (this could be a simple percentage, nothing complex with exponentials), and then figure out the monthly payment for the student.