
General course learning outcomes:

- demonstrate the use of basic programming techniques in the construction of computer programs, including techniques to collect, store, and manipulate data within a computer program.
 - apply programming techniques to solve problems in engineering.
 - complete a team programming assignment that ties together concepts learned in the class.
-

Activity 1: Inputs and Outputs - to do in lab (team)

☑ *Find unit conversion values and create computational Python programs to perform the conversions.*

As a team, write a program that makes the following five unit conversions. Ask the user for input in the first given unit, convert it to the following unit, and output the answer. Then, repeat the process for the following four unit conversions. Have your program proceed through all five conversions sequentially.

- A) Pounds to Newton
- B) BTUs to Joules
- C) Miles per Hour to Meters per Second
- D) Fahrenheit to Celsius
- E) Voltage to voltage level (power gain) measured in Decibel Volts (dBV) – for a reference of 1 volt

The activity of unit conversions might seem very basic, but it's also critically important. Conversion (a) above has a particularly costly history. Before your team leaves today, find and read a little bit about the Mars Climate Orbiter, then read these two statements from NASA:

<https://mars.nasa.gov/msp98/news/mco990930.html> and <https://mars.nasa.gov/msp98/news/mco991110.html>

Activity 2: A Very Merry Unbirthday - to do in lab (team)

☑ *Create appropriate list of variables, and plan how to create a Python program.*

☑ *Create Python program to read user input, perform necessary data reformatting, and print the expected output.*

Write a program that will read in four people's names and birthdays, and output them in formatted columns.

- A) First, as a team, create a document for submission and write a list of the variables that your team will use in this program. Your list should include:
 - a. The variable names
 - b. The type of each variable
 - c. A very brief description of what each variable is (you can write one description for multiple variables if it is clear what they all are).
- B) Next, determine what instructions you want to give to the users and write a description in your document. Determine the text you want to tell the users. Be specific about the way you want them to enter information!
- C) Third, write a description of how you want the output to be formatted. Consider how you want to align the various columns of the document (information should be lined up in clear columns).
- D) Fourth, and only after completing the above, write your code as a team. Write a Python program that asks four users for the data described, fills the appropriate variables, and outputs the result in the format you described. You should test your program to make sure it works as expected.

(continued, next page)

For parts E and F, you may need to wait until your neighboring team is ready; if so, skip to Activity 3 until that time.

- E) Once your team's code is working properly, work with a neighboring team on the following tasks:
 - a. Each team should give the other team the written instructions for running their code.
 - b. Each team should test the other's code and provide feedback (good and bad) on the direction given, how well the code worked, etc.
 - c. Each team should inspect the other's code to compare / contrast choice of variables, choice of data types, formatting, etc.
- F) Finally, your team should add to your document a summary of the feedback you received. The four of you should:
 - a. You may choose to modify your code. If you do so, add a description of what you changed to your document.
 - b. Create a short summary (probably around ½ page, and no more than 1 page), focusing on the following:
 - i. Were the names you used for the variables similar or different? Did both teams have the same sets of variables (but with different names)?
 - ii. To represent birthdays, did you use the same variable types? If both teams had the same format (same number and types of variables) for representing birthdays, then, as a team, list one other alternative that you might have used.
 - iii. Did both teams expect data to be formatted the same way – for example, was the formatting for the date the same in both pairs, and were the results printed in columns the same way?
 - iv. Were the instructions for how to format the input data clear? Think about different ways you might have asked for the birthday information to be entered.

Note: in addition to practicing reading input and formatting output, you should see that there can be different ways of constructing programs, and of assuming input might be formatted. In most programming, a programmer's choices for variables, information provided to the user, and output can vary, and still be correct.

Activity 3: Little Purple Barking Hood - to do in lab (team and individual)

- ✓ *Create an appropriate list of variables and plan for the program.*
- ✓ *Write a Python program to take user input, and format output in an organized manner.*
- ✓ *Utilize escape characters within your program.*

Building on what we learned about escape characters in strings (the `\'` and `\"` options), there are some other “escape sequences” in Python, all beginning with a `\` character. Try experimenting by printing out strings containing the following escape characters, to see what they do:

`\n, \t, \\", \b.`

If you cannot deduce what these do, feel free to look them up in an online Python guide. Make sure that *each member* of your team understands how each of these character work.

You will each write a program to generate mad-libs. A mad-lib is a short story in which a person is asked for general terms (e.g. “Number” and “Noun”), and then that word is inserted into a story. For instance, if the person entered “4” and “Giraffe”, the story might read: “I was 4 years old when I sailed a Giraffe down the Amazon River.”

- A. As a team, decide on at least 5 and no more than 10 different inputs that you will ask a user for.
 - a. Agree on these as a team (e.g., they can be things like “A person's name”, “A location”, “A color”, “Verb ending in -ing”, “Adjective”)
 - b. Agree on the order of the inputs (which will come in the story first, second, etc.)
 - c. At least one input should be a numerical value

(continued, next page)

- B. Each member should individually write a mad-lib program that forms a story using these inputs. The story should:
- Generate several sentences in a story that incorporates all of the input the user typed in.
 - Utilize all of the inputs in the correct order.
 - Present the story in a nicely formatted manner. The lines of text should be of reasonable length and flow into each other, without having single words on a line (unless at the end of a paragraph), etc.
 - Your text must make use of the following three escape characters somewhere within the text: `\n`, `\t`, and `\'`
- C. Test each other's programs.
- D. Submit all four team members programs; clearly identify which program belongs to each member.