

02 Teach: Programming Activity

ID Badge Generator

Special Instructions for this Lesson's Activity

Each lesson will contain a team activity. These activities are designed to take about one hour and should be completed by multiple people working together.

In face-to-face sections of the class, this means multiple people will sit in front of the same computer. In online sections, you need to coordinate a synchronous video call, where everyone can see the code that is being written and help contribute.

For this lesson, because the teams may not be fully defined and you have not had a chance to coordinate a meeting time yet, you are welcome to complete the activity individually or with others, making use of any synchronous or asynchronous communications you would like.

Even though you won't meet with a team this week, to help you see how the process will work, this activity is set up identically to the future team activities, including the I-Learn quiz that you will submit upon completion. **For this lesson, you should simply answer "True"** to the questions that ask if you met with your team, were on time to the meeting, etc. In future lessons, you should answer those questions honestly, according to what you actually did.

Overview

An ID badge, such as a drivers license or student ID, contains personal details that are formatted in a very specific way.

For this activity you will write a program that will create a properly formatted ID badge.

Assignment

Write a program to prompt the user for the following:

- » First name
- » Last name
- » Email Address
- » Phone Number
- » Job Title
- » ID Number

Then you should display the information back in this format:

Note that the square brackets `[]` indicate a value coming from the user and should not be displayed.

```
-----  
[LAST NAME], [first name]  
[Job title]  
ID: [id number]  
  
[email address]  
[phone number]  
-----
```

In addition to the spacing and punctuation shown above, you must implement the following requirements:

- » The last name should be converted from whatever the user types to be displayed in ALL CAPS.
- » The job title should be displayed so that the first letter is capitalized.

- » The email address should be displayed in all lowercase.

An example of the program running is shown:

```
Please enter the following information:
```

```
First name: Jane
Last name: Doe
Email address: JaneDoe@email.com
Phone number: (208) 555-1234
Job title: chief software architect
ID Number: 83-23821
```

```
The ID Card is:
```

```
-----
DOE, Jane
Chief Software Architect
ID: 83-23821

janedoe@email.com
(208) 555-1234
-----
```

I. CORE REQUIREMENTS

Each team activity will contain three core requirements. These are items that you are expected to be able to complete during the one hour team meeting if you have come prepared.

You should work through the assignment in this order:

01. Prompt for all of the necessary values and store them in variables. Then display each item to the screen, without worrying about any spacing, punctuation, or formatting yet.
02. Arrange the display so that the spacing and punctuation **exactly** match the example shown.
03. Use the built-in string functions to make the last name display in all caps, the job title display in "title case," and the email display in all lowercase.

II. STRETCH CHALLENGE

Most team activities will also contain stretch challenges, which are not specifically required by the assignment, but are a great way to dive deeper into the material. They can be more difficult and may require you to find solutions that weren't directly covered in the preparation material.

The stretch challenges for this activity are:

01. Add hair color and eye color and put them both on the same line in the display.
02. Add a field for the name of the month they started and also a yes/no field for whether they have completed advanced training. Put these both on the same line in the display.
03. For the two lines that you just added, make it so that the second columns align, regardless of how many letters are in the responses.
To complete this one, you may need to search the internet for something like, "python spacing format" or something similar to see if you get any ideas.

At the end of the stretch challenge, your output should look something like the following:

```
The ID Card is:
-----
DOE, Jane
Chief Software Architect
ID: 83-23821

janedoe@email.com
(208) 555-1234

Hair: Brown      Eyes: Blue
Month: September Training: Yes
-----
```

Sample Solution

When your program is finished, please view the sample solution for this program to compare it to your approach.

You should work to complete this activity for the one hour period first, without looking at the sample solution. However, if you have worked on it for at least an hour and are still having problems, you may feel free to use the sample solution to help you finish your program.

- » Sample solution (Core Requirements): [teach02_sample.py](#).
- » Sample solution (Stretch Challenges): [teach02_stretch_sample.py](#).

Submission

When complete, please report your progress in the associated I-Learn quiz.

If you decided to do additional work on the program after your team activity, either by yourself or with others, feel free to include that additional work when you report on your progress in I-Learn.