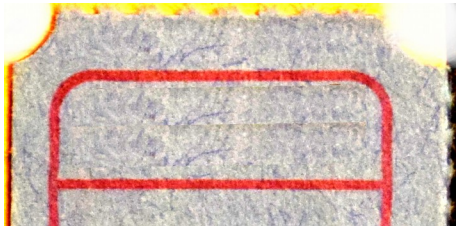
CIT-129: Python 2	*CCAC North Campus*	
 *transfer the ticket number affixed to your official grade proposal card*	**Term:** Fall 2019	**Doc Version:** v1.1
	Declared degree/certificate:	
	# Terms into program:	
	# Terms remaining:	

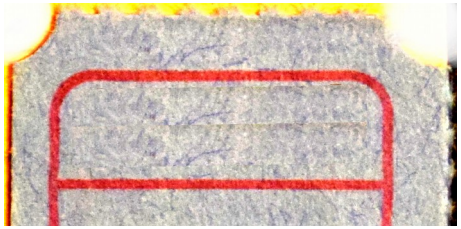
Attendance: *Using your official attendance log, compute your attendance rate*

 # days you attended class / *total # of class meetings* = *decimal percent of class attendance*

Overall attendance notes *(extenuating circumstances, prior conversations, etc.)*

Checklist Items; *Initial each item ONLY when done in the "initials" column*

#	Initials	Task description
1		**Complete** your final project. Make sure whatever code you have is working free of bugs. Note unfinished parts of your program.
2		**Post your final project work** to your **github** account by navigating into your git repository, adding your changed files, committing those changes, and pushing them.
3		**Create a readme.md** file in the same directory as your final project files on github. Use the file named "master markdown tutorial" on technologyrediscovery.net to help you with the markdown. Describe what your project does and highlight a few lines of code of which you are most proud.
4		Carefully arrive **at a fair letter grade** for your effort in the class. Write it on a 3x5 card along with a justification for WHY it is a fair grade: include in-class participation, out-of-class work, att., etc.
5		**Tackle the code review worksheet** on the back of this form as you review your peer's project and as they review yours
6	you: peer:	**Share your project with a peer** in a semi-mock-interview style format: B*e formal in your explanation of the project; use technical language; sell the hard work you did as legitimate*
7	you: peer:	**Review a peer's project** as they share it: Ask a few questions: *What did you learn doing it? Proudest parts? Improvements they would like to make?*
8	eric:	**Hand this clipboard** directly Eric when everything is done!

CIT-129: Python 2	*CCAC North Campus*	
 *transfer the ticket number affixed to your official grade proposal card*	**Term:** Fall 2019	**Doc Version:** v1.1
	Declared degree/certificate:	
	# Terms into program:	
	# Terms remaining:	

Attendance: *Using your official attendance log, compute your attendance rate*

 # days you attended class / *total # of class meetings* = *decimal percent of class attendance*

Overall attendance notes *(extenuating circumstances, prior conversations, etc.)*

Checklist Items; *Initial each item ONLY when done in the "initials" column*

#	Initials	Task description
1		**Complete** your final project. Make sure whatever code you have is working free of bugs. Note unfinished parts of your program.
2		**Post your final project work** to your **github** account by navigating into your git repository, adding your changed files, committing those changes, and pushing them.
3		**Create a readme.md** file in the same directory as your final project files on github. Use the file named "master markdown tutorial" on technologyrediscovery.net to help you with the markdown. Describe what your project does and highlight a few lines of code of which you are most proud.
4		Carefully arrive **at a fair letter grade** for your effort in the class. Write it on a 3x5 card along with a justification for WHY it is a fair grade: include in-class participation, out-of-class work, att., etc.
5		**Tackle the code review worksheet** on the back of this form as you review your peer's project and as they review yours
6	you: peer:	**Share your project with a peer** in a semi-mock-interview style format: B*e formal in your explanation of the project; use technical language; sell the hard work you did as legitimate*
7	you: peer:	**Review a peer's project** as they share it: Ask a few questions: *What did you learn doing it? Proudest parts? Improvements they would like to make?*
8	eric:	**Hand this clipboard** directly Eric when everything is done!

Python Code Review Worksheet: DAT-129 Name:

This mini-document is designed to measure your fluency in Python by asking you to analyze some lines of code in your and your peer's project.

Question 1: What makes code more or less pythonic?

At the end of our course, now is the time to reflect on the language of Python from the standpoint of "better" and "best" rather than "working" and "not working". Populate the following matrix along with a partner in class to identify more and less pythonic ways of writing.

Then, together, suggest a more python approach to a set of code and explain why it's more pythonic than the original.

Your name: _____ -

Your code: more pythonic
Your code: less pythonic
Your code: revised less pythonic code to make more pythonic:

Name of peer: _____

Peer's code: More pythonic
Peer's Code: Less pythonic
Peer's code revised: make the less pythonic code more pythonic
Explanation: What makes code "Pythonic"

Python Code Review Worksheet: DAT-129 Name:

This mini-document is designed to measure your fluency in Python by asking you to analyze some lines of code in your and your peer's project.

Question 1: What makes code more or less pythonic?

At the end of our course, now is the time to reflect on the language of Python from the standpoint of "better" and "best" rather than "working" and "not working". Populate the following matrix along with a partner in class to identify more and less pythonic ways of writing.

Then, together, suggest a more python approach to a set of code and explain why it's more pythonic than the original.

Your name: _____ -

Your code: more pythonic
Your code: less pythonic
Your code: revised less pythonic code to make more pythonic:

Name of peer: _____

Peer's code: More pythonic
Peer's Code: Less pythonic
Peer's code revised: make the less pythonic code more pythonic
Explanation: What makes code "Pythonic"