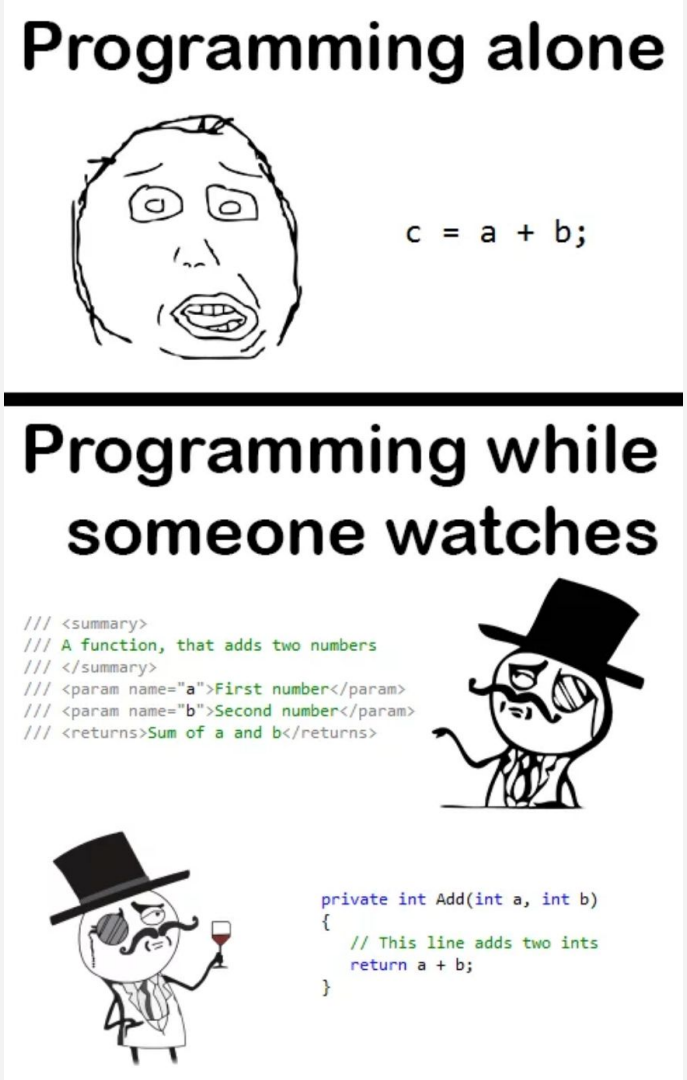
CSc 110
Quiz Example

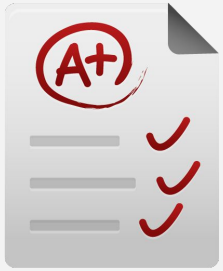
Benjamin Dicken



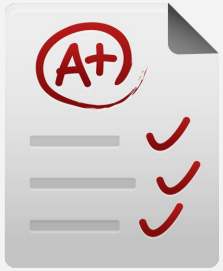
Let's build a quizzing program

- At the beginning of the program ask for:
  - The number of questions to give in the quiz (N)
  - a file to get questions/answers from

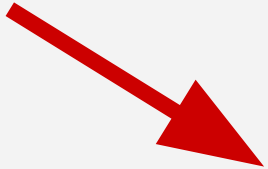
- The program will ask the user (N) questions, chosen from the questions from the input file.
  - Don't re-ask questions that user got correct.



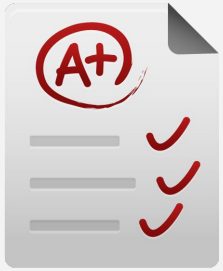
what color does the sky appear (to most people)?|||blue
how many years did it take to build the panama canal?|||10
what is the term length in years for POTUS?|||4




```
what color does the sky appear (to most people)?|||blue
how many years did it take to build the panama canal?|||10
what is the term length in years for POTUS?|||4
```

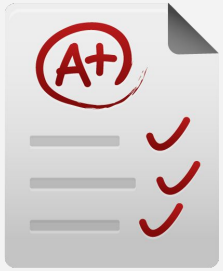


```
qa = { 'what color does the sky appear (to most people)?' : 'blue',
'how many years did it take to build the panama canal?' : '10',
'what is the term length in years for POTUS?' : '4' }
```



What are the names of the variables defined within the parentheses at a function definition?|||parameters
What data structure stores things in a sequence?|||list
True or False: Strings are immutable.|||True
What type can be used to store a number without decimals?|||int
The two types of loops we've covered are for loop and _____ loop.|||while
True or False: Every program should have a main function|||True
What symbols is used for string concatenation in python?|||+
What function can be used to get a string from a user in Python?|||input
true or false: you can have duplicate keys in a dictionary|||false
What symbols is used for multiplication in python?|||*

```
How many questions should be asked? 4
What is the name of the QA file? 110.txt
What symbols is used for string concatenation in python?
Enter response: +
correct!
What function can be used to get a string from a user in Python?
Enter response: input
correct!
true or false: you can have duplicate keys in a dictionary
Enter response: false
correct!
What symbols is used for multiplication in python?
Enter response: #
incorrect!
You got 3 out of 4 correct.
Your percentage grade: 75.0%
```



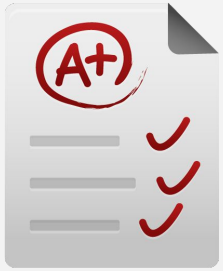
```
def load_questions_and_answers(file_name):
    # ?

def get_random_question(qa):
    # ?

def ask_question(qa):
    # ?

def main():
    # ?

main()
```



Implement load_questions_and_answers

```
def load_questions_and_answers(file_name):
    '''
    Load the questions and answers into a dictionary.

    file_name: the name of the file with the questions

    Steps:
        * Open the file
        * Create a dictionary to put the questions and answer in.
        * Add each question and answer pair into the qa dictionary.
        * return the dictionary
    '''
```

Implement load_questions_and_answers

```
def load_questions_and_answers(file_name):
    qa = {}
    file = open(file_name, 'r')
    for line in file:
        # ? ? ?
```

Implement load_questions_and_answers

```
def load_questions_and_answers(file_name):
    qa = {}
    file = open(file_name, 'r')
    for line in file:
        line = line.strip('\n')
        components = line.split('|||')
        question = components[0]
        answer = components[1]
        qa[question] = answer
    return qa
```

Implement `get_random_question`

```
def get_random_question(qa):
    '''
    Get a random question from the qa dictionary.
    qa: a dictionary, to load the questions and answers into.

    Steps:
        * Get the keys of the qa dictionary into a list
        * generate a random index
        * get one of the keys using the random index
        * return the question
    '''
```

get_random_question

```
def get_random_question(qa):
    keys = list(qa.keys())
    # ? ? ?
```

get_random_question

```
def get_random_question(qa):
    keys = list(qa.keys())
    index = random.randint(0, len(keys) - 1)
    key = keys[index]
    return key
```

Implement ask_question

```
def ask_question(qa):
    ''' Ask a question to the user.
    Select a question from the qa dictionary at random.
    qa: a dictionary, to load the questions and answers into.
    Steps:
        * Get a random question from the data
        * Ask the question
        * Get user response
        * Handle correct/incorrect response
        * Return True if correct, False if incorrect
    '''
```

Implement ask_question

```
def ask_question(qa):
    question = get_random_question(qa)
    print(question)
    response = input('Enter response: ')
    # What next?
```

Implement ask_question

```
def ask_question(qa):
    question = get_random_question(qa)
    print(question)
    response = input('Enter response: ')
    if response == qa[question]:
        print('correct!')
        del qa[question]
        return True
    else:
        print('incorrect!')
        return False
```

```
def load_questions_and_answers(file_name):
    qa = {}
    file = open(file_name, 'r')
    for line in file:
        line = line.strip('\n')
        components = line.split('|||')
        question = components[0]
        answer = components[1]
        qa[question] = answer
    return qa

def get_random_question(qa):
    keys = list(qa.keys())
    index = random.randint(0, len(keys) - 1)
    key = keys[index]
    return key
```

```
def ask_question(qa):
    question = get_random_question(qa)
    print(question)
    response = input('Enter response: ')
    if response == qa[question]:
        print('correct!')
        del qa[question]
        return True
    else:
        print('incorrect!')
        return False

def main():
    # ? ? ?

main()
```

Implement `main`

```
def main():
    ''' Outline of main:
        * Ask the user for the two input values
          (num questions and question file)
        * Load content from the file into dictionary
        * Repeatedly ask the user a random question
            * Count the correct responses
        * Report the grade, numer correct, and total
        * Recall:
            def load_questions_and_answers(qa, file_name):
            def ask_question(qa):
    '''
```

main

```
number_of_questions = int(input('How many questions should be asked? '))
file_name = input('What is the name of the QA file? ')
questions_answers = load_questions_and_answers(file_name)
correct_count = 0
for i in range(number_of_questions):
    correct = ask_question(questions_answers)
    if correct:
        correct_count += 1
print('You got', correct_count, 'out of', number_of_questions, 'correct.')
print('Your percentage grade: ' + str(correct_count / number_of_questions * 100) + '%')
```

```
import random

def load_questions_and_answers(file_name):
    qa = {}
    file = open(file_name, 'r')
    for line in file:
        line = line.strip('\n')
        components = line.split('|||')
        question = components[0]
        answer = components[1]
        qa[question] = answer
    return qa

def get_random_question(qa):
    keys = list(qa.keys())
    index = random.randint(0, len(keys) - 1)
    key = keys[index]
    return key

def ask_question(qa):
    question = get_random_question(qa)
    print(question)
    response = input('Enter response: ')
    if response == qa[question]:
        print('correct!')
        del qa[question]
        return True
    else:
        print('incorrect!')
        return False

def main():
    number_of_questions = int(input('How many questions should be asked? '))
    file_name = input('What is the name of the QA file? ')
    questions_answers = load_questions_and_answers(file_name)
    correct_count = 0
    for i in range(number_of_questions):
        correct = ask_question(questions_answers)
        if correct:
            correct_count += 1
    print('You got', correct_count, 'out of', number_of_questions, 'correct.')
    print('Your percentage grade: ' +
      str(correct_count / number_of_questions * 100) + '%')

main()
```