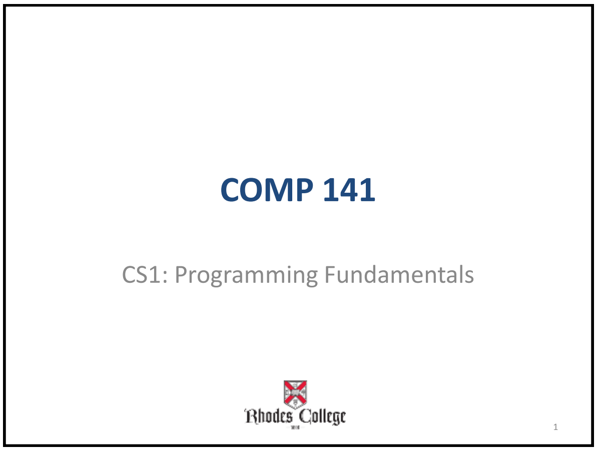
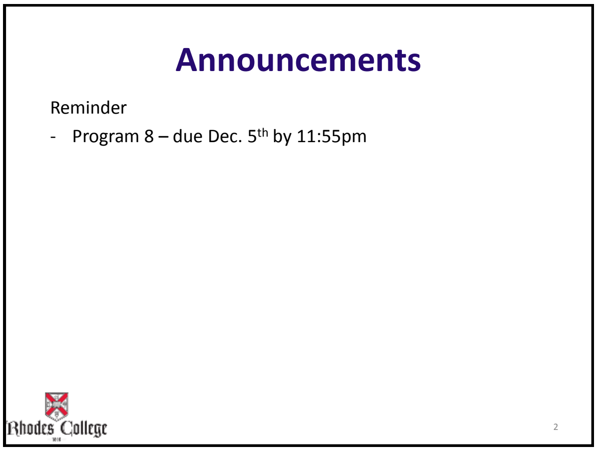


COMP 141

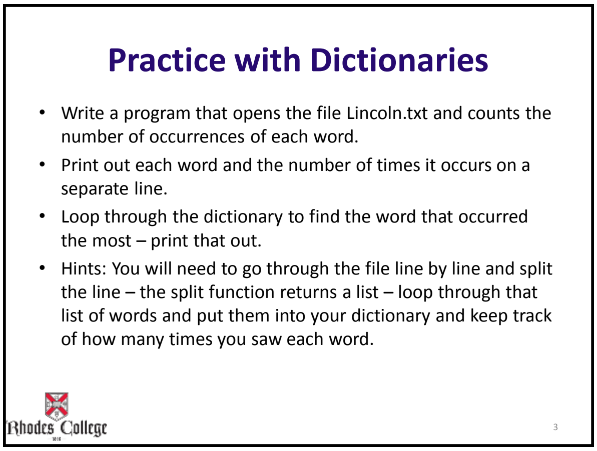
CS1: Programming Fundamentals



Announcements

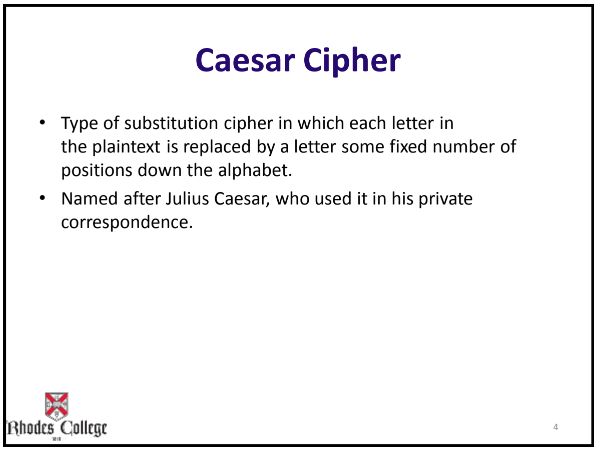
Reminder

- Program 8 – due Dec. 5th by 11:55pm



Practice with Dictionaries

- Write a program that opens the file Lincoln.txt and counts the number of occurrences of each word.
- Print out each word and the number of times it occurs on a separate line.
- Loop through the dictionary to find the word that occurred the most – print that out.
- Hints: You will need to go through the file line by line and split the line – the split function returns a list – loop through that list of words and put them into your dictionary and keep track of how many times you saw each word.



Caesar Cipher

- Type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- Named after Julius Caesar, who used it in his private correspondence.

Example of a Caesar Cipher

```
 Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ Cipher:
XYZABCDEFGHIJKLMNOPQRSTUVW
```

- Transformation can be represented by aligning two alphabets.
- This one uses a left rotation of three places, equivalent to a right shift of 23

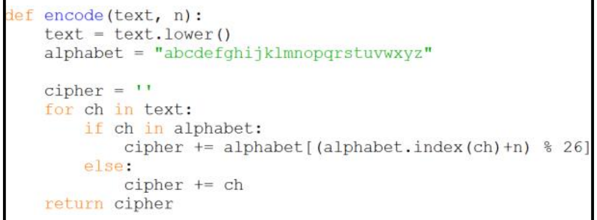
Encrypting a Message

```
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
Plaintext: the quick brown fox jumps over the lazy dog
```

- Look up each letter of the message in the "plain" line and write down the corresponding letter in the "cipher" line.
- The encryption can also be represented using modular arithmetic
  - Transform the letters into numbers, according to the scheme, A = 0, B = 1,..., Z = 25.
  - Encryption of a letter by a shift *n* can be described mathematically as,

$$E_n(x) = (x + n)\ \%\ 26$$

Encrypting



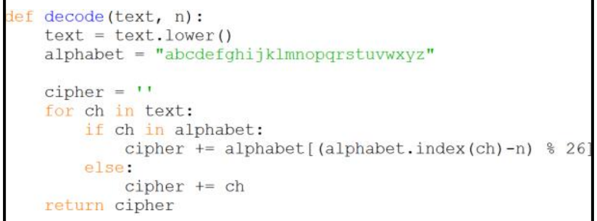
```
def encode(text, n):
    text = text.lower()
    alphabet = "abcdefghijklmnopqrstuvwxyz"

    cipher = ''
    for ch in text:
        if ch in alphabet:
            cipher += alphabet[(alphabet.index(ch)+n) % 26]
        else:
            cipher += ch
    return cipher
```

Decrypting a Message

- Opposite process of encryption
  - Look up each letter of the message in the “cipher" line and write down the corresponding letter in the “plain" line.
- Can also use modular arithmetic to solve

$$\mathrm{D}_n(x) = (x - n)\ \%\ 26$$

Decrypting



```
def decode(text, n):
    text = text.lower()
    alphabet = "abcdefghijklmnopqrstuvwxyz"

    cipher = ''
    for ch in text:
        if ch in alphabet:
            cipher += alphabet[(alphabet.index(ch)-n) % 26]
        else:
            cipher += ch
    return cipher
```

Breaking the Cipher

- If you were not the intended recipient of the message, you wouldn’t be told the shift (n).
- To determine what n is:
  - Analyze the frequency of each character in the encrypted message
  - Since ‘E’ is the most common letter used in English, you should find the shift from E to the most common letter in the message

Practice

- Write a Caesar cipher program to decode the message in cipherText.txt (in Box.com folder).
- You will need to read in the file (you can read it in all as 1 line if you’d like).
- Get the frequency count for each letter (don’t include spaces).
- Find the shift between the max letter and ‘e’
- Use the shift to decode the message