

# JAVA PROGRAMMING

## (340)

### REGIONAL – 2014

*TOTAL POINTS* \_\_\_\_\_ (*340 points*)

**Failure to adhere to any of the following rules will result in disqualification:**

- 1. Contestant must hand in this test booklet and all printouts. Failure to do so will result in disqualification.**
- 2. No equipment, supplies, or materials other than those specified for this event are allowed in the testing area. No previous BPA tests and/or sample tests or facsimile (handwritten, photocopied, or keyed) are allowed in the testing area.**
- 3. Electronic devices will be monitored according to ACT standards.**

No more than 90 minutes testing time

Property of Business Professionals of America.  
May be reproduced only for use in the Business Professionals of America  
*Workplace Skills Assessment Program* competition.

**Note to Contestant:**

The rubric for this event grants points for each minor task completed. That being the case, the contestant should strive to complete as many of the rubric items as possible. The contestant should have received a flash drive from the proctor. A text file, cipher.txt, must be stored on the flash when you receive it.

Create a folder named with your contestant ID number (ex. xx-xxxx-xxxx). Make certain all of the files associated with your program are contained in the folder and that the program will compile/run from the folder. Failure to do so will result in the loss of points under the “Application/Execution” rubric section. Make sure you use this contestant ID number when you are commenting your code, instead of your name.

## Decipher my Ciphertext

### Description:

In the language of cryptography, *ciphertext* refers to a message encoded with a particular *key*. *Plaintext* refers to the original, unencoded text. In this problem, both the ciphertext and the key are simply strings of upper-case characters. The ciphertext is generated from the plaintext by “adding” corresponding characters of the plaintext and the key together. If the plaintext is shorter than the key, only some of the key will be used. Similarly, if the plaintext is shorter than the key, the key will be used multiple times.

For example, to encode the plaintext “HELLO” with the key “CAT”:

Plaintext: HELLO  
Key: CATCA  
Ciphertext: KFFOP

And to encode the plaintext “DOG” with the key “FIDO”:

Plaintext: DOG  
Key: FID  
Ciphertext: JXX

To add two letters together, use the following convention: A=1, B=2, ..., Z=26. If the sum of two letters is greater than 26, subtract 26 from the sum.

For example:  $A + E = 1 + 5 = 6 = F$ , and  $D + X = 4 + 24 = 28 = 2 = B$ .

Given a ciphertext/key pair, determine the corresponding plaintext.

### Input:

The input file (ciphers.txt) will consist of pairs of lines, with the first line being the ciphertext and the second line being the key. Both will consist of only uppercase letters.

### Output:

For each ciphertext/key pair, print the original ciphertext/key pair and the corresponding plaintext message to the screen and also write the same output to the file “plain.txt”.

### Example file:

#### Input (from ciphers.txt):

KFFOP  
CAT  
JXX  
FIDO

#### Output (to plain.txt):

KFFOP/CAT = HELLO  
JXX/FIDO = DOG

The file names for both the input and output file are entered on the command line. If no file names are entered, display "Usage Error: Not enough Arguments."

Your application will be graded on the following criteria:

<b>Application/Execution</b>	
<ul style="list-style-type: none"> <li>Application reads the names of the files from the command line</li> </ul>	_____ 30 pts
<ul style="list-style-type: none"> <li>Application reads from the input text file</li> </ul>	_____ 30 pts
<ul style="list-style-type: none"> <li>Application correctly displays to the screen the original ciphertext/key pair and the corresponding plain text</li> </ul>	_____ 40 pts
<ul style="list-style-type: none"> <li>Application correctly converts the ciphertext into plain text</li> </ul>	_____ 60 pts
<ul style="list-style-type: none"> <li>Application reads the entire file (ciphers.txt)</li> </ul>	_____ 30 pts
<ul style="list-style-type: none"> <li>Application correctly writes the ciphertext/key pair and corresponding plain text to the output text file (plain.txt)</li> </ul>	_____ 40 pts
<ul style="list-style-type: none"> <li>Application does not display any extra results from reading past the file</li> </ul>	_____ 20 pts
<ul style="list-style-type: none"> <li>Application does not write any extra results to the "plain.txt" file</li> </ul>	_____ 20 pts
I/O error handling is done if files cannot be opened	_____ 10 pts
I/O error handling is done if no command arguments	_____ 10 pts
JavaDoc comments are used	_____ 10 pts
Code uses a class for the conversion	_____ 20 pts
Methods are commented	_____ 10 pts
Code copied to USB drive and program runs from USB	_____ 10 pts

Total Points: \_\_\_\_\_ 340 pts