**public** **class** CaeserCipher {

// note: fields are declared private

**private** **static** String *alphabet*; // fields that live inside an Object also call instance variable

**private** **static** String *shiftedAlphabet*; // data that are encapsulated in an Object, it is created when new is called

**private** **static** **int** *mainKey*;

**public** CaeserCipher(**int** key){ // constructor - obj can be used using new

*mainKey* = key;

*alphabet* = "ABCDEFGHIJKLMNOPQURSTUVWXYZ";

*shiftedAlphabet* = *alphabet*.substring(key) +

*alphabet*.substring(0, key); // initialize using key

}

**public** **static** String encrypt(String input){

StringBuilder sb = **new** StringBuilder(input);

**for** (**int** i = 0; i < sb.length(); i++){

**char** c = sb.charAt(i);

**int** idx = *alphabet*.indexOf(c);

**if** (idx != -1){

c = *shiftedAlphabet*.charAt(idx);

sb.setCharAt(i, c);

}

}

**return** sb.toString();

}

**public** **static** **int** [] countLetters(String message) {

**int** [] counters = **new** **int** [26];

**for** (**int** i = 0; i < 26; i++) {

counters[i] = 0;

}

System.***out***.println(counters);

String alph = "abcdefghijklmnopqrstuvwxyz";

message = message.toLowerCase();

**for** (**int** i = 0; i < message.length(); i++) {

**int** position = alph.indexOf(message.charAt(i));

**if** (position != -1) counters[position] += 1;

}

**return** counters;

}

**public** **static** **int** maxIndex(**int** [] counters) {

**int** maxcounter = 0;

**int** maxposition = 0;

**for** (**int** i = 0; i < counters.length; i++) {

**if** (maxcounter < counters[i]) {

maxposition = i;

maxcounter = counters[i];

}

}

**return** maxposition;

}

**public** **static** String decrypt(String message) {

// CaeserCipher cc = new CaeserCipher(17);

// int[] freqs = countLetters(message);

// int maxDex = maxIndex(freqs);

// int dkey = maxDex-4;

// if (maxDex < 4)

// dkey = 26 - (4 - maxDex);

CaeserCipher cc = **new** CaeserCipher(26-*mainKey*);

**return** cc.*encrypt*(message);

}

**public** **static** **void** testdecrypt() {

CaeserCipher cc = **new** CaeserCipher(17);// initialize by passing 17 inside the new object

String message = *encrypt*("FIRST LEGION ATTACK EAST FLANK!");

System.***out***.println(message);

String decrypt = *decrypt*(message);

System.***out***.println(decrypt);

}

**public** **static** **void** main (String[]args){

//CaesarCipher cs = new CaesarCipher();

*testdecrypt*();

}}

Output:

VYIJK BUWYED UKKUSA UUJK VBUDA!

EHURS KZFHNM ZSSZBJ ZZRS EKZMJ!