COMP125 Semester 2 2014
Assignment 1 – Remaining Details

August 25, 2014

Extension

Some students may appreciate having some time over the weekend to finalise their work. Also this sheet advising remaining details comes to you later than originally planned (with our apologies). So we offer every student a 48 hour extension to the assignment deadline. The new deadline is **11 pm on Sunday 7 September**.

Detailed marking scheme

As previously advised, your work will be marked out of 24, and is worth 6% of your final mark for this unit.

The machine tests for the basic methods – that is, the four methods `isNumeric`, `checkDigit`, `encrypt` and `cardSecCode` – will be marked out of 12. That is, up to 3 marks will be awarded for the results of machine testing of each method.

Visual inspection of the basic methods plus `main` will be marked out of 8. For each method `checkDigit`, `encrypt` and `cardSecCode` you will receive up to 2 marks for code quality. Each method `isNumeric` and `main` could earn you 1 mark for code quality. As previously advised, the marker will be looking for clear, simple code, good programming style, and adherence to the specifications stated.

The relatively advanced tasks on decryption set out below will be marked out of 4. Machine testing of `decrypt` will be worth 2 marks. Your answers in comment form to the questions asked below concerning decryption will be marked out of 2.

Description of advanced tasks (for 4 marks out of 24)

As previously discussed, encryption is an essential part of the CSC algorithm. Encryption is also important in other security features of credit cards and other kinds of accounts.

Encryption is however of limited use without its companion *decryption*. Decrypting a cryptogram is the process of obtaining the original card number using the secret keys which were used for the encryption.

The purpose of the relatively advanced tasks is to stimulate you to investigate the process of decryption. In particular consider the following questions:

1. For which keys k_1 and k_2 having nonnegative integer values less than 10 could we be assured that decryption of encrypted card numbers will always be possible? Try to describe in words the properties of such keys. Provide a concise answer as a comment in your submitted file.

2. For such keys k_1 and k_2, how could we perform decryption? Provide a decryption rule $D(c) = \cdots$, where c denotes the integer value of a digit character of a cryptogram, and $D(c)$ denotes the integer value of the decrypted character. Your decryption rule must satisfy $D(E(d)) = d$ for all integers d. (That is, given integer d, decrypting the encryption of d results in d.) Provide a concise answer as a comment in your submitted file.

3. Implement a method for decryption based on your decryption rule:

```
/**
* Returns the decryption of the given cryptogram using the given keys.
* Precondition: it can be assumed that cryptogram is a valid numeric string,
*               k1 is a valid first encryption key,
*               k2 is a valid second encryption key.
* Postcondition: the string returned is the decryption of cryptogram
*                using keys k1 and k2, according to the rule outlined
*                in the comments above.
*/

public static String decrypt(String cryptogram, int k1, int k2)
```

Hints and pointers for basic tasks

Here are some hints some of which have already been mentioned in lectures:

1. For coding some of the methods it may be helpful to observe that the character `ch` is a decimal digit character exactly when `'0' <= ch && ch <= '9'`.

2. To obtain the integer value of the decimal digit character `digitChar` you could use

```
int digitValue = (int) digitChar - (int) '0';
```

3. To convert a decimal string into hexadecimal (base 16) notation there is a Java method you could use if you wish. Look at methods of the class `Long`.