
Due Date:	By 11:59pm Wednesday October 12, 2016
Evaluation:	3% of final mark (see marking rubric at the end of handout)
Late Submission:	none accepted
Purpose:	The purpose of this assignment is to help you learn the Java selection and flow of control statements, <code>if</code> , <code>if/else</code> , <code>while</code> and <code>do/while</code> loops.
CEAB/CIPS Attributes:	Design/Problem analysis/Communication Skills

General Guidelines When Writing Programs:

Please refer to the handout of Assignment #1.

RECOMMENDATION for this Assignment:

Take the time to write out an algorithm for these problems to help clarify the logic BEFORE you start programming. (No need to hand in the algorithm).

Question 1: User integer input & selection

A company that transmits data over the telephone is concerned that its phone might be tapped. All data is transmitted as four-digit `int` values. You have been hired by **JuniorProgrammers-R-Us** to write a Java application that encrypts its data so that the data may be transmitted more securely. Encryption is the process of transforming data into a form that can be recognized only by the intended receiver. Your application should read a single four-digit integer input by the user and encrypt the information as described in the following steps.



<http://www.cloudpro.co.uk/cloud-essentials/cloud-security/2011/achard-newden-and-the-rise-of-cloud-encryption/page/0/>

Your application should:

- Display a welcome message
- Prompt the user for a four-digit number to encrypt and read it into an `int` variable.
- If the number entered is less than 4 digits (< 1000) or greater than 4 digits (> 9999) then send the user an error message indicating what the problem is and end the program, if not move on to step d)
- Extract each digit and save them into four `int` variables. (Hint: integer division and the modulus might be useful here).
- Encrypts each digit by replacing it with the (sum of that digit plus 7) modulo 10. Swap the 1st digit with the 3rd and the 2nd digit with the 4th.
- Display the encrypted version of the number with a descriptive message.
- Display a closing message and end the program.

Following are a few sample output screens to illustrate the expected behavior of your application. (Text in green is user input)

```
Nancy's 4-digit Encrypter
-----
Please enter the four digit number you wish to encrypt:
1234

The encrypted version of your input is 0189

Thank you for using Nancy's 4-digit Encrypter program.
```

Figure 1 – Valid input

```
-----  
Nancy's 4-digit Encrypter  
-----  
  
Please enter the four digit number you wish to encrypt:  
12345  
Sorry, but that is not a 4-digit number. Program will terminate.  
  
Thank you for using Nancy's 4-digit Encrypter program.
```

Figure 2 – input more than 4-digits

```
-----  
Nancy's 4-digit Encrypter  
-----  
  
Please enter the four digit number you wish to encrypt:  
123  
Sorry, but that is not a 4-digit number. Program will terminate.  
  
Thank you for using Nancy's 4-digit Encrypter program.
```

Figure 3 – Input less than 4-digits

Question 2: User string input, selection and while loop

<http://donwebblog.com/2012/10/las-peores-contrasenas-del-ano/>

JuniorProgrammers-R-U is so happy with your code for question 1, that they now have asked you to write another Java application to generate passwords. The program should prompt the user for a 5 characters string. It then reverses the order of the string and subtracts 7 from each character's ASCII code to generate the password. If the user enters a string that does not contain exactly five characters, the application should display an error message and exit.

Your application should:

- Display a welcome message
- Prompt the user for a 5- characters string and read it into a single String variable.
- If the string is not exactly 5 characters long send an error message and proceed to step f), if not continue on to step d).
- Using a while loop, generate the password in a new String variable by reversing the order of the characters in the string and subtracting 7 from each character to produce the password.
- Display the password with a descriptive message.
- Display a closing message and end the program.

```
-----  
Nancy's Password Generator  
-----  
  
Please enter a 5 character word which will be used to generate a password:  
abc45  
Your password is .-\[Z  
  
Thank you for using Nancy's Password Generator program.
```

Figure 4 – Valid input (lowercase letters and digits)

```
-----  
Nancy's Password Generator  
-----  
  
Please enter a 5 character word which will be used to generate a password:  
HaPpy  
Your password is riIZA  
  
Thank you for using Nancy's Password Generator program.
```

Figure 5 – Valid input (mix of lowercase and upper case letters)

```
-----  
Nancy's Password Generator  
-----  
  
Please enter a 5 character word which will be used to generate a password:  
Na_9u  
Your password is n2XZG  
  
Thank you for using Nancy's Password Generator program.
```

Figure 6 – Valid input (mix of lowercase and upper case letters as well as characters)

```
-----  
Nancy's Password Generator  
-----  
  
Please enter a 5 character word which will be used to generate a password:  
1234  
Sorry, but that is not a 5-character string. Program will terminate.  
  
Thank you for using Nancy's Password Generator program
```

Figure 7 – Input less than 5-characters

Question #3: Selection Statements & Looping (do/while)

DrawAnElephant is a game (based on the British game Beetle [https://en.wikipedia.org/wiki/Beetle_\(game\)](https://en.wikipedia.org/wiki/Beetle_(game))) in which one draws an elephant in parts. The game is based on random die rolls, with no skills involved.

The part drawn are decided on the roll of a die. The rolls are as follows:

- 6 is for an ear, of which there are two.
- 5 is for a leg of which there are four.

- 4 is for a tail, of which there is one.
- 3 is for the body, of which there is one.
- 2 is for the head, of which there is one. (The head contains 2 eyes and a mouth).
- 1 is for the trunk, of which there is one.

You need to have a body before you can draw any other parts. To the body you may add the head, legs and tail. You need to have a head before you can draw the trunk and the ears. The first player to draw all of the requisite pieces wins the game.

Your job is to write a Java application which simulates the **DrawAnElephant** game. Here is a sample output screen to illustrate the expected behavior of your application.

Note: To simulate the throwing of the die, consider using the `random()` function in the `Math` class.

`Math.random()` returns a double random number greater 0.0 and less than 1.0.

Here are a few examples of the use of `random()`.

`int num1 = (int)(Math.random() * 5);` will store an integer between 0 and 4 inclusive in the variable `num1`.

`int num2 = (int)(Math.random() * 5) + 1;` will store an integer between 1 and 5 inclusive in the variable `num2`.



<https://elephantspokenhere.wordpress.com/2015/12/16/what-a-wonderful-story-wendi-the-elephant-inspires-christmas-fostering-at-the-david-sheldrick-wildlife-trust-with-endearing-adopt-me-video/>

Nancy's Elephant Drawing Game

Status after round 1:

Player 1 rolled a 1 while Player 2 rolled a 4.

Player 1's elephant has no body, no head, no ear(s), no trunk, no tail, and no leg(s).

Player 2's elephant has no body, no head, no ear(s), no trunk, no tail, and no leg(s).

Status after round 2:

Player 1 rolled a 6 while Player 2 rolled a 5.

Player 1's elephant has no body, no head, no ear(s), no trunk, no tail, and no leg(s).

Player 2's elephant has no body, no head, no ear(s), no trunk, no tail, and no leg(s).

.

Status after round 5:

Player 1 rolled a 6 while Player 2 rolled a 3.

Player 1's elephant has no body, no head, no ear(s), no trunk, no tail, and no leg(s).

Player 2's elephant has a body, no head, no ear(s), no trunk, no tail, and no leg(s).

Status after round 6:

Player 1 rolled a 2 while Player 2 rolled a 5.

Player 1's elephant has no body, no head, no ear(s), no trunk, no tail, and no leg(s).

Player 2's elephant has a body, no head, no ear(s), no trunk, no tail, and 1 leg(s).

Status after round 7:

Player 1 rolled a 3 while Player 2 rolled a 4.

Player 1's elephant has a body, no head, no ear(s), no trunk, no tail, and no leg(s).

Player 2's elephant has a body, no head, no ear(s), no trunk, a tail, and 1 leg(s).

.

Status after round 12:

Player 1 rolled a 6 while Player 2 rolled a 6.

Player 1's elephant has a body, a head, 1 ear(s), no trunk, no tail, and 2 leg(s).

Player 2's elephant has a body, a head, 2 ear(s), no trunk, a tail, and 2 leg(s).

Status after round 13:

Player 1 rolled a 5 while Player 2 rolled a 2.

Player 1's elephant has a body, a head, 1 ear(s), no trunk, no tail, and 3 leg(s).

Player 2's elephant has a body, a head, 2 ear(s), no trunk, a tail, and 2 leg(s).

Player 1 rolled a 3 while Player 2 rolled a 5.

Player 1's elephant has a body, a head, 2 ear(s), a trunk, a tail, and 3 leg(s).

Player 2's elephant has a body, a head, 2 ear(s), a trunk, a tail, and 4 leg(s).

>>>> Congratulations to player 2!!! Your elephant is complete! <<<<

Hope you enjoyed drawing elephants!

Submitting Assignment 2

- Zip the source code (the .java file only please) of this assignment.
- Naming convention for zip file: Create one zip file, containing the source files for your assignment using the following naming convention:
The zip file should be called *a#_studentID*, where # is the number of the assignment and *studentID* is your student ID number.
For example, for the second assignment, student 123456 would submit a zip file named *a2_123456.zip*
- For submission instructions please refer to the course web page of your section, as it may not be the same for each section/instructor.
- **Assignments not submitted to the correct location or not in the requested format will not be graded.**

Evaluation Criteria for Assignment 2

For Assignment 2 – 5 pts.	
Comments - description of variables/ description of the steps in code/ purpose of program	1 pts.
Choice of variable names	1 pt.
Indentation and readability of program	1 pt.
Welcome and closing message	1 pt.
Clear prompts to user & clear messages with output	1 pt.

Question 1 – 5 pts.	
Input from keyboard	1 pt.
Extracting digits	1.5 pts.
Forming encrypted number	1.5 pts.
Handling of error cases	1 pts.
Question 2 – 5 pts.	
Input from keyboard	1 pt.
Extracting each character	1 pt.
Converting each character and generating password	2 pts.
Handling of error cases	1 pt.
Question 3 – 5 pts.	
Drawing elephant in correct order (body before head, head before trunk, etc.)	2.5 pts.
Determining winner in drawing game	2.5 pts.