University of Victoria
Department of Computer Science
CSC 110 Fundamentals of Programming
ASSIGNMENT 2

DUE: January 29, 2018 before 7:50 am – By submission on ConneX only

Assignment Instructions

How to hand in your work

Submit the `Totem.java` and `PandI.java` files that fully answers parts (c) and (d) (below) through the Assignment #2 link on the CSC 110 conneX site under Assignments. Please make sure you follow all the required steps for submission (including the final confirmation of your submission).

Objectives:

Once you have successfully completed this assignment, you should be able to:

- use a for loop in a Java program.
- write a Java 'static method'.
- build-up a more complex program from simple methods.
- generate random numbers in a specified range.
- Input from the keyboard and use that input in a program.
- Use if statement to chose from a menu.
- carefully follow the Java coding conventions, as posted for the course.

Part (a): Problems from the Textbook

Complete the Chapter 2&3 Self--Check Problems and compare your answers to those given by the textbook authors at: http://www.buildingjavaprograms.com/self-check-solutions-4ed.html

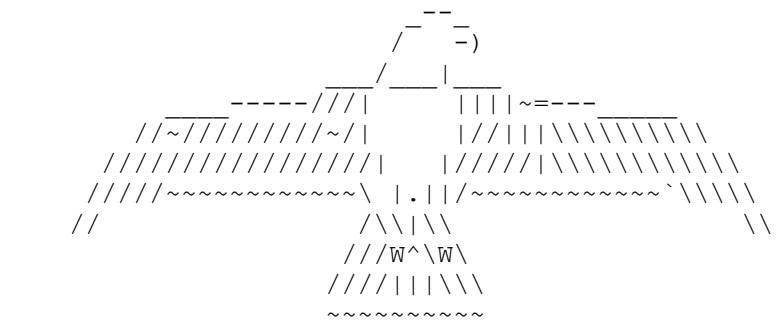
Part (b): Introduction, methods and Tester Program

The history of the West Coast totems goes back for generations. They are carved from tall cedar trees. Each contains multiple figures. Traditional figures include; ravens, bears, whales and (of course) people. There a number of totems on campus, including one near front of the Engineering Office Wing, which was a created by artist Charles Elliot.

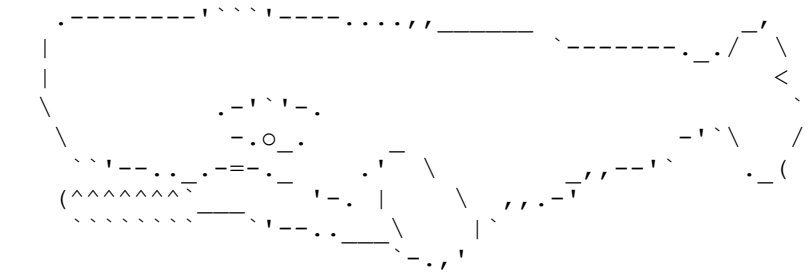
You might wonder what this has to do with programming... It turns out that individual entities are combined in different ways within different totems, and THAT has a structural similarity to what happens with "methods" that are executed one after another in programs. We hope to make that structural relationship clear in this assignment!

This assignment again uses ASCII art, this time to simulate a totem that consists of eagle, whale and human figures. The simulated figures are stylized in our system as follows:

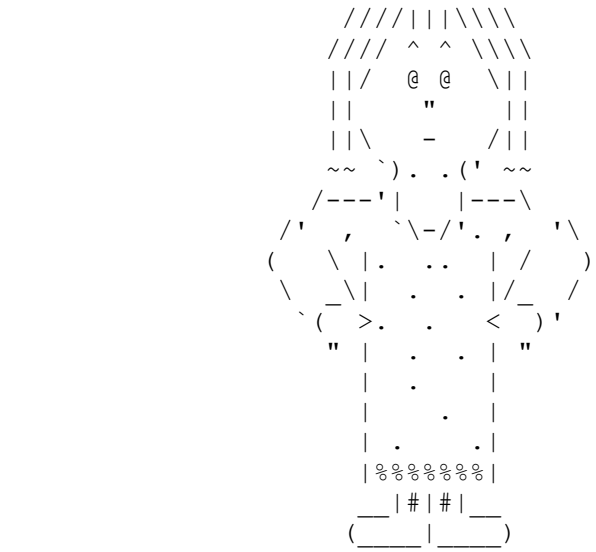
Eagle:



Whale:



Human:



- In a file called Totem.java, write and test three static methods, one to output each of the stylized figures that will be on the totems. The method signatures for those methods are as follows:

```
public static void eagle()
public static void whale()
public static void human()
```

- Test the methods by writing a main that calls all three and confirms to you that they produce exactly the figures above. (This is called a 'tester' program because it simply tests methods.) It might look like this:

```
public static void main(String [] args) {
      eagle();
      whale();
      human();
}
```

- Document each static method, as described in the coding conventions document (found on the course ConneX site in Resources.)

Part (c): Full Totem Program

Write a Java program that uses the three methods written in Part (b) above, in particular, it does the following:

→ Outputs a title,

→ Asks the user to specify the number of figures (`numFigs`),

→ Does the following `numFigs` times:

- Use a random number generator and if statements to choose one of the three figures (1-eagle, 2-whale, 3-human), as follows:

```
//place immediately below your new Scanner line:
Random rand = new Random();

// . .

// Use if & random to choose a figure
int choice = rnd.nextInt(3)+1;
// Choose the figure
if (choice == 1) eagle();
if (choice == 2) whale();
if (choice == 3) human();
```

→ Document this program, as described in the coding conventions document (found on the course ConneX site in Resources.) Give special consideration to the conventions for spacing and documentation of methods.

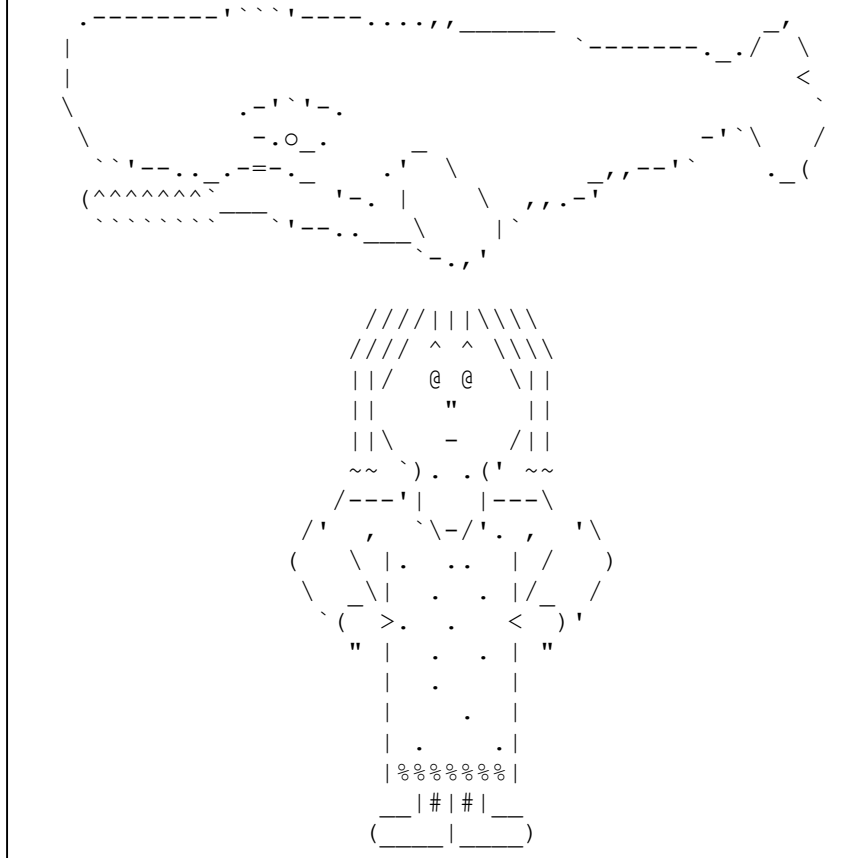
File to submit: **`Totem.java`**

See next page for sample output

Sample output of this program:

```
T O T E M   G E N E R A T O R

BY: LillAnne Jackson
   January 16, 2018
How many creatures? ==> 2
```



Part (d): Calculating Daily Interest

In a separate file, called `PandI.java`, write a program that calculates the amount of interest charged on a bank loan. Here are the specifics:

- → Output a title and concise directions.
- → Input the original loaned principal from the user.
- → Input the annual interest rate from the user.
- → Assuming that the interest rate is annual, but compounded daily over the year, divide the interest rate by the number of days in a year.
- → Use a loop to calculate the principal plus interest each day for 1 year.
- → Output the initial principal, the total amount payable at the end of the year and the calculated interest.

Document this program, as described in the coding conventions document (found on the course ConneX site in Resources.)

Sample output of this program:

```
I N T E R E S T   C A L C U L A T O R
BY: LillAnne Jackson
   January 16, 2018
INPUT:
  Amount of Loan (in $)==> 100
  Annual Interest (in %)==> 5
Original Principal = 100.00
Amount Owing (1 year later) = 105.13
Interest Cost = 5.13
```

NOTE: For the purposes of this assignment numbers like `5.13` and `5.13000000000` and `5.129999999` will be considered equal. In particular, there is no need to restrain your floating point numbers to only 2 places after the decimal point.

File to submit: **`PandI.java`**

Grading: In addition to checking that both programs compile and run (with correct output), The markers will be looking specifically for the following: 1) Header, Class and Method-header comments in both programs; 2) all identifier names appropriate (according to the coding conventions) 3) the Totem uses the provided Random & if statements in a for loop; 4) The PandI program uses a for loop to calculate the principal plus interest (not Math.pow).