# CMPS 4143

Programming Assignment-2 (Due October 11th 11:59 PM)

[*To get the full credit of your programming assignment, find the problem solving steps and documentation guidelines listed in 'Programming Assignments' section in D2L. You are supposed to submit a zip file (don't use winrar to zip) that should contain a documentation file like a docx file and your codes with proper programming extensions like .java or .py. Make sure all of your code is compliable or no runtime error.TA will not grade you if your code is not runnable and get a zero*]

1. Write an OOP code (A case study: Should be unique; If you are copying code from online you will be caught) that has the following OOP features: inheritance(any), polymorphism (runtime and compile time), abstraction and encapsulation (20 points)

2. Consider the following organigram (can be found in https://en.wikipedia.org/wiki/Mammal) of Mammalia family. (a: 5, b: 5, c: 10, d: 20, e:10) – (50 points)
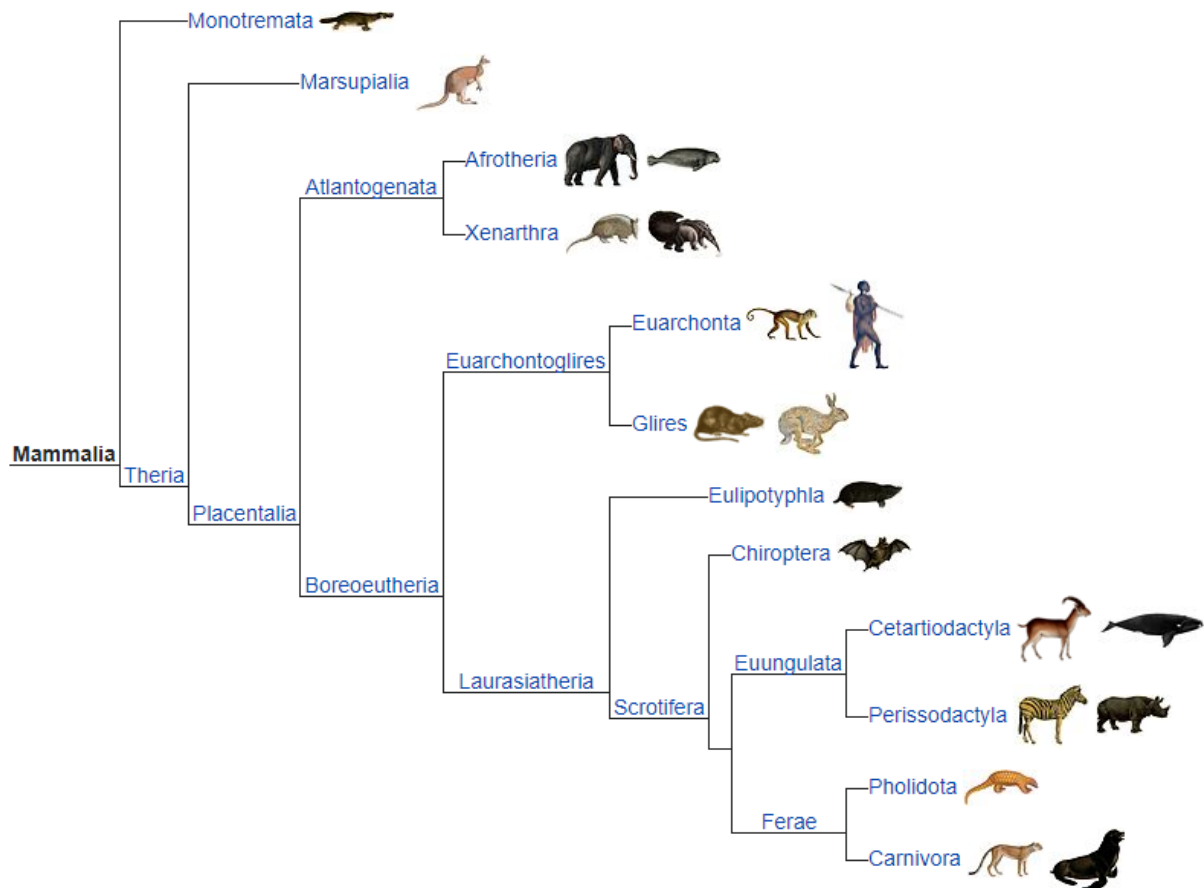
    a. Each of members
        i. has
            1. Legs (count)
            2. Tails (yes/no)
            3. Brain (yes/no)
            4. Neurons (count)
            5. Precision to understand (0.00 to 100.00%)
            6. Birth date
            7. Favorite foods
        ii. can
            1. Eat
            2. Walk
            3. Sleep
            4. Give Birth

b. Think about perissodactyla, they can mate with carnivora members; can breed and produce a new family called 'Perivora'. Perivora animals supposed to give birth to a new family 'Periveron'. But by nature law no periveron member can be crated.

c. Create at least 5 animals from each family; Perviveron member cannot be produced.

d. Now you can do some creativity by changing the neurons, precision to understand, food habits, but you can't change the legs, brains, birth date. Change these features for at least 20 animals and show how it was before and how it is now. From level 6 to 8(Considering Mammlia as level-1 and carnivora, cetartiodactyla are on level-8), animals can run and hunt for their food. Change at least 10 animals' running speed, hunting process and hunting food type for those who are 6-8 level animals. Try changing at least 10 animals and show (reading from files; see next bullet) what they were and what they are now.

e. All outcomes should be stored in a file. When you have created objects, their information should be stored in files. After changing their behaviors and attributes, they should also be stored in files. The term show mentioned in the previous sub-section refers to reading the data from files and printing to the console.

3. Create a calculator class which can perform certain operations: addition, subtraction, multiplication, division, modulo, and power calculation. Now, add Java exception handling to your code by defining two exception classes `SyntaxError` and `RuntimeError`. A `SyntaxError` exception should be thrown when an illegal character is found, a closing `)` is not found, or a `=` is not used in a let expression. A `RuntimeError` exception should be thrown when an identifier is encountered for which no value can be found. Include in the file documentation a list of each type of exception that your program throws. The exceptions should propagate the error to the main program which prints the diagnostics of the error. You must handle these errors using Java exceptions and the message should be printed by a Java exception handler in a catch clause. (30 points)

Compile, run, and test your program with at least these expressions:

```
Test Expression                 Correct Response
---------------                 ----------------
1+(2*3;                         syntax error: ')' expected
(let x 5) + x;                  syntax error: '=' expected
(let x = 5) (let y = 6);        syntax error: operator expected
(let x = 5 let y = 6);          syntax error: ')' expected
(ler x = 5) ^ (let y = 6);      runtime error: 'ler' undefined
(let x = 5) + y;                runtime error: 'y' undefined
```