COMP 2500 – Object Oriented Programming 1

Lab 23/3/2015

1. Complete the *Mammal* class with the following requirements:
   1. *Mammal* is abstract.
   2. It contains one variable *name*.
   3. The constructor takes in a value for *name* and sets the *name* attribute accordingly.
   4. Contains getters and setters for *name*.
   5. Has one abstract method:

*public void eat(String foodItem)*

Assume the implementation of *eat* prints messages as follows:

“I am eating hay.”

1. Complete the *Elephant* class with the following requirements:
   1. *Elephant* is a subclass of *Mammal*.
   2. Contains two instance variables, *age* (*int*) and *isSleeping* (*boolean*).
   3. Contains a suitable constructor.
   4. Contains getters and setters for its variables.
   5. A *moveTruck()* method which takes in no parameters and prints a message such as the following:

“Tim is moving his trunk.”

* 1. A *toString()* method which prints the *name* and the *age* of the elephant.

1. Create an exception called *CannotEatException*.
2. Make alterations to your *eat* method in your *Mammal* and/or *Elephant* class(es) to incorporate the *CannotEatException*.
3. Write code snippets for your *Test* class which creates a *Mammal* object and calls the *eat* method for the *Mammal*. Your code should take into account the *CannotEatException*, which can be generated.
4. Determine if the following statements will compile and run successfully. Run the relevant code snippet from the *Test* class to test your answers.
   1. Elephant e = new Elephant("Ele", 4);

Mammal m = e;

m.eat("peanuts");

m.moveTrunk();

* 1. Elephant e = new Mammal("Timmy");

e.eat("nuts");

e.moveTrunk();

1. Determine the output of the following statements. Run the relevant code snippet from the *Test* class to test your answers.
   1. Elephant e2 = new Elephant("Phino", 3);

System.out.println(e2.toString());

* 1. Mammal m = new Elephant("Li", 3);

System.out.println(m.toString());