CS1302
Exam01 Review

PLAS:

Xander Aviles and (Aiden) Geonhui Lee

To be used in supplement with:

```
$ git clone git@github.com:xanaviles/cs1302-exam01Review.git
```

Navigating Unix

1. Change into cs1302-exam01Review and do not leave unless directed otherwise (cd)
2. Look around the directory (ignore the jar file for now) (ls and find)
3. Create src/cs1302/practice and src/cs1302/inter in one command and without changing directories (mkdir and pipes)
4. Move Moveable.java into src/cs1302/interface and move the rest of the java files into src/cs1302/practice (mv)
5. Create *cs1302.practice.Cow* by copying Bear.java to the new file, Cow.java (mv & cp)
6. [bonus] Change into the src/cs1302/practice directory, how would you change back into cs1302-exam01Review in one command? (using "..")

Directory Result:

cs1302-exam01Review

- bin // don't forget to make bin
- src
  - cs1302
    - inter
      - Moveable.java
    - practice
      - Animal.java
      - Mammal.java
      - Fish. java
      - Cow.java
      - Bear.java

- **Commands not mentioned but should be reviewed:**

  grep, echo, chmod (permissions, octal, …), which, stat, pwd,

Dependencies & UML

(Interfaces and Abstract Classes | Reference and Object Types)

1. Use cat to determine dependencies of the classes.

File Name	Type of Class	Depends On
Moveable.java	Interface	
Animal.java	Abstract	Moveable
Mammal.java	Abstract	Moveable, Animal
Cow.java	Concrete	Moveable, Animal, Mammal
Bear.java	Concrete	Moveable, Animal, Mammal
Fish.java	Concrete	Moveable, Animal

2. Draw the UML for these relationships:

   answer can be found here

   *remember you implement Interfaces and extend Abstract classes
3. Adjust your Cow.java.
4. Compile these classes in the correct order. What errors did you get? Why? and How would you fix them? (overriding methods, implementing abstract methods)
5. Create the class and directory for cs1302.drivers.umlDriver
6. Paste this code into umlDriver.java

Exception Handling

Javadoc

For this example we will only be writing Javadoc comments for Bear.java

1. Write a javadoc comment for all of the methods in Bear.

Nodes

1. Create cs1302.

NodeDriver.java

```
//properly import listadt

//properly specify package

public class NodeDriver {
        //be aware of 3 Node constructors

        // 1. Node()
        // 2. Node(String str)
        // 3. Node(String str, Node next)

        //Block 1
         StringList.Node a = new StringList.Node();
        a.setName(“1”);

        //Block 2
        a.setNext(new StringList.Node(“2”));
        a.getNext().setNext(new StringList.Node(“3”));

        //Block 3
        StringList.Node b = new StringList.Node(“0”, a);

        //Block 4
        b.getNext().getNext().setNext(“5”);

        //Block 5
        b.getNext().setNext(b.getNext().getNext().getNext());
}
```