

## CSE5910 : Multimedia Programming in Java

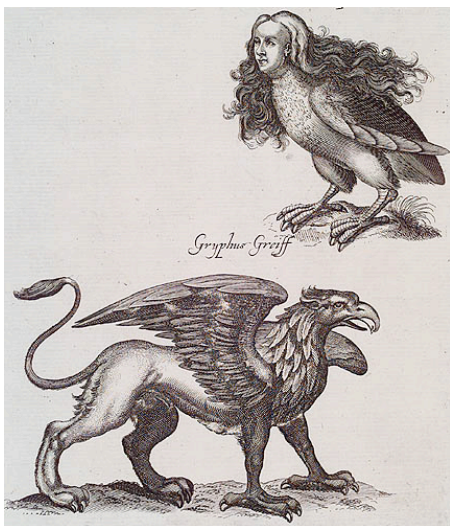
Laboratory Session Worksheet. Week 2, Semester 2, 2007

### Preliminary Organisation.

1. Organise yourselves into assignment groups of **four** students if you have not already done so. (It is not compulsory for you to form groups based on your laboratory session members, but it may make things more convenient for you. Exchange email address and telephone numbers so that you can stay in touch with one another.
2. Select the host student and ensure that they have the student ID numbers and authcate *us-ernames* of all students in your group.
3. Organise a regular meeting time and place that your whole group can meet to discuss the assignment. Enter these details into your diaries!

### Individual Exercises.

1. Type in, compile and execute the "Hello World" program from the first week's lecture using the Linux OS and the command line. Make sure you understand the process.



2. Design a simple Java class `Beast` that represents mythological beasts or monsters with these fields:

- i. Its number of legs.
- ii. Whether it has wings.
- iii. The number of eyes it has.
- iv. Its name.
- v. Its favourite food.
- vi. Its scariness.

3. Write a class `Zoo` that instantiates a few `Beasts` such as an Ogre, a Harpie, a Centaur, a Dragon and a Hydra or some other different monsters of your own choosing (do some quick research).

4. Print out the details of your `Beasts` from within the `Zoo` using a `ListAttributes()` method in the class `Beast` and a `ListBeasts()` method that you add to the class

`Zoo`.

5. Add a method to the class `Beast` called `Frightens()`. This method must return a Boolean value and take as a parameter an object of type `Beasts`. It will return `true` if the `Beast` calling the method frightens the `Beast` passed as a parameter. Otherwise it will return `false`. Use the scariness of the `Beast` to determine which `Beast` frightens which other `Beast`.
6. Read up on the Java archive tool `jar`. Convert your Java `.class` files into an executable `jar` file.

**Tip:** You'll need to ensure that a file (call it `JavaMainClass` for example) exists with a single line in it telling the JVM where to find your `main()` function when it executes the `jar` file or you'll get an error message: `Failed to load Main-Class manifest attribute from MyJarFile.jar`

## CSE5910 : Multimedia Programming in Java

This file must contain the line: `Main-Class: Zoo` followed by a carriage return if that is the file that has the `main()` method in it.

To have the archive tool include this information in the jar file use the `-m` option like this:

```
jar -cmf JavaMainClass MyArchive.jar *.class
```

7. Investigate the Java class `Scanner`. Write a simple text-based system that allows users to interactively: (i) choose a Beast from the Zoo for the software to describe; (ii) test on a case-by-case basis which Beasts frighten which other Beasts in your Zoo. Make the system as simple to use as you can.
8. Investigate some existing pieces of software e.g. (i) a word processor (ii) the Unix shell (iii) a drawing program (iv) another piece of software of your choice. What methods are available for selecting objects? Write a list of these and comment on the advantages and disadvantages of each.