CSE 1325-001
Dinosaur Bones
Course Project Phase 2
Assigned: February 17, 2016
Due Week of March 21st
Demonstrations in class or during office hours
Last day to turn in: March 25th

Congratulations! You have reached Phase 2 of the CSE 1325 semester project. Now, all of you have experience with Java. As such, it is time to move further into Java and Object Oriented Programming.

1 Penalties for Phase 2.

Certain types and features must be implemented in phase 2 of the project.

1.1 You must have two programs and three packages.

For this project, the program must still have a buyer program and a seller program. The Dinosaur Bone Seller Tool and Dinosaur Bone Buyer Tool remain separate entities. In addition, now the class is further along with Java, the project must contain three *packages*. One package should be the seller tool, called "**seller**". One package will be the buyer tool, called "**buyer**". And the third package will be all the data types. The map, map grid points, and all the Dinosaur Bones and continents. This package will be called "**datastore**"

1.2 The project must have two diagrams.

Continuing from Phase 1, the class diagram will have to be updated to meet the requirements of Phase 2. As before, all the classes in the class diagram have to match your code. Starting with Phase 2, it is required that the constructors also be included in the class diagram. (see below). In addition, generalization or ***inheritance*** must be included in the class diagrams.

1.3 The project must have comments.

Comments will be required for this project. In particular, each file should include the name of the class, the authors, the authors' student id numbers, and a short description of the classes in that file. Each class should have a set of comments including the name of the class, and any data members defined in the scope (namespace) of the class. Each function should include the function name, a description of its parameters, and a description of its return values. If Input and Output happen within the function, these should be defined.

1.4 All objects will have constructors that will set their beginning values.

Constructers need to be used for all user defined classes in the project. In particular, the Coordinate Class, Dinosaur Bones, buyer, and seller, and grid points should all be created using a constructor that contains their key data values. (Such as longitude and latitude). For those with inheritance structure, the keyword **super** should be used to initialize the basic data fields.

1.5 Try/Catch blocks now must be around all input/output statements.

For all user interfaces and file interfaces, try/catch blocks will now be required. This means file input, file output and keyboard input. Try and catch blocks are not required for systems outputting the text map and similar routines to the console screen.

1.6 Inheritance is a must in Phase 2 of the project.

The project must exhibit factors and features of inheritance. These features of object oriented programming must be in the programs for this project and their accompanying diagrams**.**

2 New Features for the Programs

2.1 Buyer Class required in Buyer Tool

Create a class called Buyer. Buyer should contain a Coordinate, and a name. The Buyer class should have a menu to allow a buyer to be added, to be deleted, or to be updated. The Buyer must also be saved to file.

2.2 Add Seller Class to Seller Tool

Create a class called Seller. Seller should contain a Coordinate, and a name. The Seller class should have a menu to allow a seller to be added, to be deleted, or to be updated. The Seller must also be saved to file.

2.3 Add Bone in one line of text

Add the ability to add a Dinosaur Bone to your system by inputting a single line of text.

For example:

0.00,0.00,Spinosaurus,50000.00

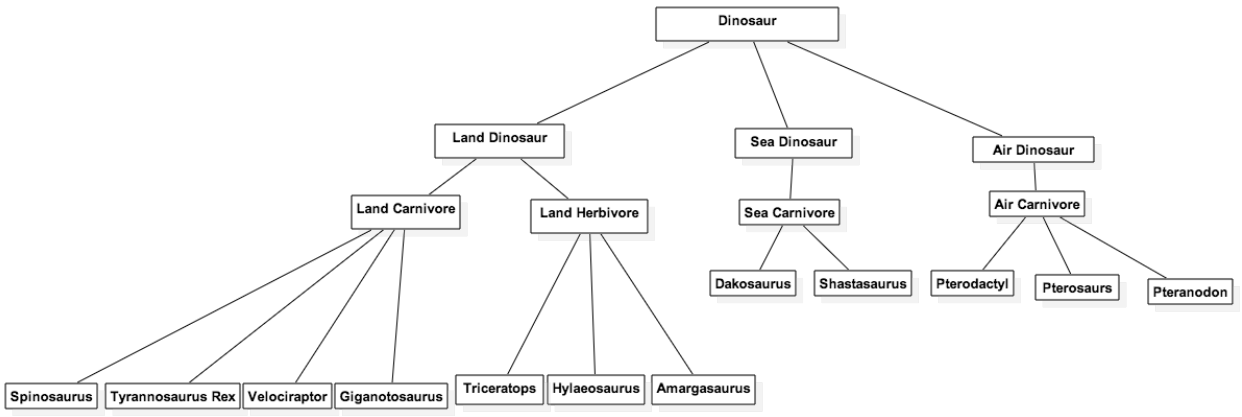
Would add a Dinosaur Bone at 0 longitude, 0 latitude, of type Spinosaurus and it costs $50000.00.

2.4 Implement a Coordinate Class

Many of the items in these programs rely on the longitude and latitude of the object in action. For this system of programs, include a class called Coordinate. Coordinate has four data members: longitude as a double, latitude as a double, rowIndex as an integer, and colIndex as an integer. Coordinate also must be visible on your Class Diagrams.

3 Inheritance: More Dinosaur Bones

There are hundreds of different classifications of Dinosaur Bones, all depending on their types of habitat and diet. Create an inheritance structure that has the following types of Dinosaur Bones



3.1 Class Dinosaur Bone

This is the superclass of all Dinosaur Bones. This class should have a data member of type Coordinate, a price, and an id of some kind. Mainly, the dinosaur bone from Phase 1.

3.1.1 Land Dinosaur

All land dinosaurs have an additional attribute, a double, for speed. The default speed will be 15.0 mph.

3.1.1.1 Land Carnivore

All land carnivores will have an additional feature, ground speed, which indicates how fast they can chase down their prey.

3.1.1.1.1 Spinosaurus

Spinosaurus is famous for being bigger than Tyrannosaurus Rex, famous for having a giant spiny sailfin on its back. Spinosaurus should have an integer attribute for the number of spines in its sailfin.

3.1.1.1.2 Tyrannosaurus Rex

The Tyrannosaurus Rex is the long beloved monster of Hollywood, posters in children's bedrooms, and always the star of the show. Tyrannosaurus Rex class will have the additional attribute "smelling range" due to the creature's enormous nasal cavity. In addition, a method should generate a random number of the range of 1 through 5 to indicate how scary the Tyrannosaurs Rex is.

3.1.1.1.3 Velociraptor

The Velociraptor is a fast-moving carnivorous dinosaur, usually depicted as a stalking, wolf-pack style animal that surprises its prey with speed. In reality, the Velociraptor has actually three key sizes, small, medium, and large. Include an integer attribute that can represent the size of the velociraptor. In addition, and a method that returns not the integer, but the label "small" "medium" or "large".

3.1.1.1.4 Giganotosaurus

The Giganotosaurus is the second largest predator of the dinosaurs, larger than T. Rex, and smaller than Spinosaurus. As a result, Giganotosaurus is the unwanted predator. Giganotosaurus should have a method called poorLittleMe(), when called, will print out to the screen "Bigger than T. Rex, but not as cool."

3.1.1.2 Land Herbivore

Land herbivore's tend to be depicted as walking on four feet, and as a result, they should have a measurement attribute called gait, which is a double.

3.1.1.2.1 Triceratops

The triceratops is a favorite of humans, shown on everything from movies to cereal boxes. As a result, everyone wants one. There is an added cost of $2500 for being so lucky as to have a Triceratops bone. This should be done as an attribute of type double.

3.1.1.2.2 Hylaeosaurus

The Hylaeosaurus is an armored land herbivore dinosaur, with three spiked armored plates on each shoulder. An additional attribute for this dinosaur will be armor thickness on the left, and armor thickness on the right. This values for both left and right should be randomly generated in its constructor on a range from 1 to 3.

3.1.1.2.3 Amargasaurus

The Amargasaurus is a land herbivore dinosaur with a ridge of spikes down its back. Recently, a punk rock band has started calling itself "Amargasaurus Rex" has been releasing tunes, and on the cover they have their hair in a spiked mohawk while wearing a crown. Anyone buying a piece of Amargasaurus Bone gets a free CD of their music and a download code. The download code is a sixteen character code that is an attribute of the Amargasaurus class.

3.1.2 Sea Dinosaur

All sea-going dinosaurs should have a boolean to indicate if they are fresh water or salt water.

3.1.2.1 Sea Carnivore

All sea going carnivores should have an integer and a method to describe their propulsion: flippers, tails, or feet. An integer to determine the type, and a method to return the relevant label. It could be more than one type of propulsion.

3.1.2.1.1 Dakosaurus

The Dakosaurus is a sea carnivore not unlike to a crocodile, capable of snatching pterodactyls out of the air for a snack. With four limbs, this swimming monster is known for its bite. As a result, companies selling Dakosaurus bones are also selling boogie-boards with a fake bite out of them saying “I survived the Dako!”. Add a method to the Dakosaurus class to print out “I survived the Dako!” when called.

3.1.2.1.2 Shastasaurus

The Shastasaurus is possibly the largest sea going reptile in the fossil record. Surprisingly, this huge creature is known to have no teeth. The shastasaurus should always have an integer describing the number of teeth being always zero. Use a static attribute.

3.1.3 Air Dinosaur

All air dinosaurs have wings, and as a result, they should have a double attribute to represent the wingspan of the creature.

3.1.3.1 Air Carnivore'

All air carnivores will have to have teeth. There is something inherently strange of thinking of flying creatures that have teeth. All air carnivores have an integer to indicate how many teeth they have.

3.1.3.1.1 Pterodactyl

Pterodactyls are flying predator dinosaurs the size of a large bird at most. These creatures tend to be on the small side. They should have an additional method called "carryACoconut" and have a boolean attribute to indicate if the Pterodactyl is carrying a coconut.

3.1.3.1.2 Pterosaurs

Pterosaurs are flying predator dinosaurs the size of a small single-prop airplane. These creatures tend to have large wingspans and to have a colored crest atop their head. Every Pterosaur should have a string attribute to describe the crest. For example, the crest could be "red with white polka-dots" or "green with yellow stripes".

3.1.3.1.3 Pteranodon

The Pteranodon is a huge flying dinosaur, with a wingspan of easily 20 feet or more. These large winged beasts take to the sky and cast a huge shadow over the landscape. Sellers marketing Pteranodon bones have an additional market: fantasy Pteranadon saddles. The Pterandon bones have two attriubtes, a boolean for the availability of a saddle, and a double for the cost of a saddle. Basic price for a saddle is $1000. In addition, the Pteranadon has a method called "specialOffer" which will add the cost of the fantasy saddle to the dinosaur bone.

4 New Class: The Continents

Included this time is a set of continent files. northamerica.txt, southamerica.txt, austrailia.txt, asia.txt, europe.txt, and antartica.txt. Each is a set of rows and columns that represent the continent. Each file is mutually exclusive. Create a continent class that will contain a two dimensional Boolean array. Have a member function that takes a class Coordinate and returns true if the Dinosaur Bone is on that particular continent.

Continent would also include a price increase data member for calculating the price of the Dinosaur Bone. Use the following table to assign the price increase.

Continent	Price Increase
North America	+1000
South America	+3000
Europe	+1000
Africa	+3000
Asia	+4000
Antarctica	+100000
Australia	+5000

5 Dinosaur Bone Price Function

Create an abstract function for Dinosaur Bones called “pricing” that will automatically generate a base price of the Dinosaur Bone when it is created. This would be presented to the operator of the program when creating a new Dinosaur Bone (after longitude, latitude, and type are selected) The operator then has the option of overriding the suggested Dinosaur Bone price. Cost will be a combined function of the kind of Dinosaur Bone and the location of a Dinosaur Bone.

6 Scramble!

Have a menu option that will completely scramble the map using the Random functions in Java. Have all the Coordinate classes reset to new values Buyer, Seller, and all types of Dinosaur Bones are scrambled and put back on the map. Dinosaur Bones must be on ***land***.

7 Bonus Work (20 points)

Find out the distance, in miles, between a Buyer and a Seller given they have Coordinates on the globe. This is not as easy as it sounds. An east west direction is 360 degrees, while north south is 180 degrees. Given that the Earth is an oblate spheroid, that means that a degree in longitude is not exactly the same value as a degree in latitude. Or is it? Research the mathematical solution, and write a function to find the distance between two points on a globe in miles given longitude and latitude. This function must be called from the main menu of the Seller tool and include Buyer information.