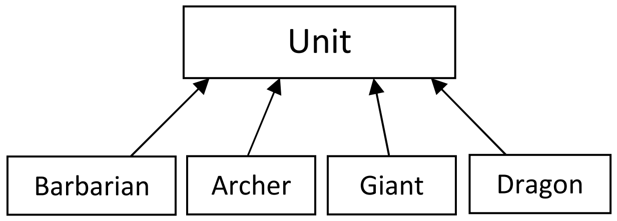
Unit 5 – WS06 Interfaces

Name ____

The goal of this assignment is to create one interface and four classes that implement the interface in such a way that the following inheritance hierarchy is represented (think Clash of Clans on this one):



Problem #1 – Create the ***Unit*** interface. The interface should contain the following list of methods:

1. ***getLevel*** method that returns an integer level.
2. ***getType*** method that returns the unit type.
3. ***getHP*** method that returns integer hit points.
4. ***attackStatus*** method that returns true or false.
5. ***levelUp*** method that increments unit level by 1.

Problem #2 – Create the ***Barbarian, Archer, Giant, and Dragon*** classes that implement the Unit interface.

Problem #3 – Add constructor methods to the four classes that implement the Unit interface.

a. All units should contain four instance variables: type, level, hp, and attackStatus.
b. When constructed, units receive level 1 status, are given their unit type, have an attackStatus set to false, and are assigned hit points according to the following chart
(L=level and HP=hit points):

	L	HP		L	HP
Barb.	1	45	Giant	1	300
	2	54		2	360
	3	65		3	430
Archer	1	20	Dragon	1	1900
	2	23		2	2100
	3	28		3	2300

Problem #4 – Write the 5 methods for each of the four units so that the Unit interface is implemented in each.

a. ***getLevel( )*** returns the object's level.
b. ***getType( )*** returns the object's type (String).
c. ***getHP( )*** returns the object's hp.
d. ***getAttackStatus*** returns true or false.
e. ***levelUp( )*** should add 1 to the object's level.

Problem #5 – Edit levelUp( ) for all five parts of this program so that the following occur:

a. Current unit level is required as a parameter.
b. If the current unit level is 1 or 2, increment the unit's level and reassign the new hp (see chart)
c. If the current unit level is 3 (highest allowed in this game) give output stating that the unit is already at its max level and no upgrade is made.

Problem #6 – Create a unitMain class that contains a main method. Experiment, explore, and test all aspects of the previous classes in this worksheet (i.e. build various units, adjust levels & hp, verify each method is working, etc.).

****

Review Challenge –

a. Create yet another class called ***UnitBuilder***.
b. Review how to create an ArrayList:
List<**type**> **listName** = new ArrayList<**type**>( );
c. Build 4 empty ArrayLists called barbarianList, archerList, giantList, and dragonList (of type Barbarian, Archer, Giant, and Dragon). This actually creates lists for each unit type. When completed you will have the ability to have an army comprised of multiple of each unit type.
d. Create 4 methods in the UnitBuilder class called buildBarbarian, buildArcher, buildGiant, and buildDragon. Each of these methods should construct a new instance of each type of unit and add the newly constructed instance to the appropriate ArrayList.
e. Create a unitBuilderMain class to test the following:

```
unitBuilder build = new unitBuilder();
build.newBarbarian();
build.newBarbarian();
build.newArcher();
for(int i=1;i<=5;i++)
   build.newDragon();
System.out.println("# Barbarians: "+build.barbarianList.size());
System.out.println("# Archers: "+build.archerList.size());
System.out.println("# Giants: "+build.giantList.size());
System.out.println("# Dragons: "+build.dragonList.size());
```

OUTPUTS:

```
# Barbarians: 2
# Archers: 1
# Giants: 0
# Dragons: 5
```