

COMP 2150 – Spring 2014

Homework 2: Classes and Objects

(Posted Feb. 5 – Due Feb. 12 by 12:40 pm)

Remember that as stated on the syllabus, this assignment should be an individual effort (contact me or visit the Computer Science Learning Center, www.cs.memphis.edu/cslc, if you need help).

Submission: Please zip your source code into a single file (you can zip the entire project folder if you're using BlueJ) and upload it to the proper folder in the eCourseware dropbox at <https://elearn.memphis.edu>. The dropbox will cut off all submissions after the indicated deadline, so don't wait until the very last minute to submit your work!

Grading: This assignment will be graded by the TA Rahul Vemuri (rvemuri@memphis.edu). If you have questions or concerns about your grade, please contact him first. I'll be happy to look over your assignment myself if he's not able to resolve the situation to your satisfaction. Also remember that as stated on the syllabus, all submissions MUST compile and run to receive credit. The TA does not have time to find and correct your syntax errors!

Coding Style: Be sure to follow the good coding practices that were discussed in COMP 1900:

- Use descriptive variable and method names. Avoid using single-character names unless it's very obvious what the variable's being used for (like a loop counter).
- Follow standard Java programming conventions for **variableAndMethodNames**, **ClassNames**, **CONSTANT_NAMES**.
- Consistently indent your code.
- Use comments judiciously. Every source code file should include a comment block at the top that lists your name and the assignment number. Every non-trivial method should have a comment preceding it that specifies what the method does, what its parameters are, and what it returns. For simple methods like accessors or mutators this isn't necessary, but it doesn't hurt to put in a simple comment like `// accessor methods`. I also expect to see comments throughout your code explaining what actions are being taken!

NOTE: You will be extending your code from this assignment in Homework 3, so this would not be a good assignment to skip!

Suppose you're designing a video game where the objective is to travel around the world collecting cute cartoonish creatures that become stronger over time by fighting other cute cartoonish creatures. (This is totally the best idea ever! Why hasn't anyone thought of this before??!!)



1. **(20 pts)** Write a class **CuteCreature**, such that each **CuteCreature** object has the following instance variables:
 - Species
 - Level – a positive integer that indicates how powerful the creature is
 - Current hit points – a measure of how much damage the creature can take before being incapacitated
 - Maximum hit points
 - Attack damage – a measure of how much damage this creature inflicts when it attacks another creature
 - Experience points – a measure of how close the creature is to “leveling up” and becoming more powerful
 - Experience value – how many experience points this creature is worth when defeated by another creature
 - isSpecial – this is a **boolean** variable that determines whether this creature is considered “special.” Internally, special creatures are exactly the same as non-special creatures. However, they have a slightly different appearance and are somehow very highly prized by the players of your game.

CuteCreature should include at least the methods below. Feel free to add other methods (such as getters/setters) if you want.

- A constructor that allows you to specify the species, maximum hit points, attack damage, experience value, and “special” status when a **CuteCreature** object is first created. A newly created **CuteCreature** should have a level of 1, current hit points equal to the maximum hit points, and zero experience points.

- **public void takeDamage(int dmg)**

This method allows the **CuteCreature** to take the specified amount of damage to its current hit points. Assume that negative hit points are not allowed, so this method should ensure that the current hit points cannot go below zero.

This method should also display some text when called, to indicate the amount of damage taken. If the damage is enough to bring the current hit points to zero, display some text to indicate that the **CuteCreature** has been incapacitated.

- **private void levelUp()**

This method allows the **CuteCreature** to “level up.” Leveling up increases the creature’s level by 1, in addition to the following changes:

- If the new level is between 2-10 (inclusive): Current and maximum hit points increase by 4, attack damage increases by 3
- If the new level is 11 or over: Current and maximum hit points increase by 1, attack damage increases by 1
- For all level ups: Experience value increases by 10

Note that **levelUp** is declared **private** since it’s meant to be called only from this class’s **gainExp** method below.

This method should also display some text when called, to indicate that the creature is leveling up.

- **public void gainExp(int exp)**

This method allows the **CuteCreature** to gain the specified number of experience points, leveling up if necessary. Note that more than one level may be gained from a single call of this method, if a sufficiently large argument is provided! Assume that 250 experience points are required to advance from level 1 to level 2, and the experience per level increases by 50 for each level thereafter. To illustrate how this works, here’s a chart of the experience needed for the first several levels:

Current Level	Experience Points Needed for Next Level	Total Experience Points Accumulated at Next Level
1	250	250
2	300	550
3	350	900
4	400	1300
5	450	1750

This method should also display some text when called, to indicate the amount of experience gained.

- **public void attack(CuteCreature c)**

This method allows the calling **CuteCreature** to make a single attack against another **CuteCreature**. Assume the attacking creature has a 75% chance to hit with the attack, a 5% chance to score a “critical hit,” and a 20% chance of missing altogether. If it hits, the attack damages the target creature by a random amount within $\pm 20\%$ of the attack damage of the attacking creature. So for example, if the attacking creature has an

attack damage of 10, a hit could do anywhere between 8-12 damage. A critical hit doubles the usual damage. If it misses, nothing happens. If the target creature is defeated (i.e, has its current hit points brought down to zero) by the attack, the method should also make the calling creature gain the appropriate amount of experience.

This method should also display some text when called, to indicate which creature is attacking which and the results of that attack.

- A **toString** method that shows all the instance variables of the **CuteCreature**. For example, if you make a new **CuteCreature** with the code

```
CuteCreature c = new CuteCreature("Bowlbasore", 40, 5, 70, true);
```

then calling

```
System.out.println(c);
```

might result in something like

```
Level 1 Bowlbasore
-----
*** Special! ***
HP: 40/40
Attack Dmg: 5
XP: 0/250
XP Value: 70
```

2. (5 pts) Write a client program that instantiates a number of **CuteCreature** objects. Thoroughly test your methods to ensure that your code is the very best, that no code ever was.

Below is an example of what you might see when you run your tests.

```
Level 1 Bowlbasore
-----
HP: 40/40
Attack Dmg: 6
XP: 0/250
XP Value: 600

Level 1 Skwer-Tell
-----
*** Special! ***
HP: 40/40
Attack Dmg: 6
XP: 0/250
XP Value: 600

Bowlbasore attacks Skwer-Tell!
Hit! Skwer-Tell took 7 damage!

Skwer-Tell attacks Bowlbasore!
Miss!

Bowlbasore attacks Skwer-Tell!
Hit! Skwer-Tell took 7 damage!
```

Skwer-Tell attacks Bowlbasore!
Critical hit! Bowlbasore took 14 damage!

Bowlbasore attacks Skwer-Tell!
Hit! Skwer-Tell took 6 damage!

Skwer-Tell attacks Bowlbasore!
Hit! Bowlbasore took 7 damage!

Bowlbasore attacks Skwer-Tell!
Hit! Skwer-Tell took 7 damage!

Skwer-Tell attacks Bowlbasore!
Hit! Bowlbasore took 5 damage!

Bowlbasore attacks Skwer-Tell!
Hit! Skwer-Tell took 5 damage!

Skwer-Tell attacks Bowlbasore!
Miss!

Bowlbasore attacks Skwer-Tell!
Hit! Skwer-Tell took 6 damage!

Skwer-Tell attacks Bowlbasore!
Hit! Bowlbasore took 6 damage!

Bowlbasore attacks Skwer-Tell!
Critical hit! Skwer-Tell took 10 damage!
Skwer-Tell fainted!
Bowlbasore defeated Skwer-Tell!
Bowlbasore gained 600 experience!
Bowlbasore leveled to 2!
Bowlbasore leveled to 3!

Level 3 Bowlbasore

HP: 16/48
Attack Dmg: 12
XP: 600/900
XP Value: 620

Level 1 Skwer-Tell

*** Special! ***
HP: 0/40
Attack Dmg: 6
XP: 0/250
XP Value: 600