

# ITEC 120: Principles of Computer Science I

## Homework 3 – Monster testing

>>>> **Due Date: Friday, February 7th 10:00PM via Desire 2 Learn** <<<<<<

In doing this homework, remember to abide by the RU Honor Code.

---

### Problem 1

5 points

The World of Wyvern testing program you created impressed your superiors. From the data you provided, the expansion pack team decided that the monster is deadly to newer players, but is reasonable to include in group quests.

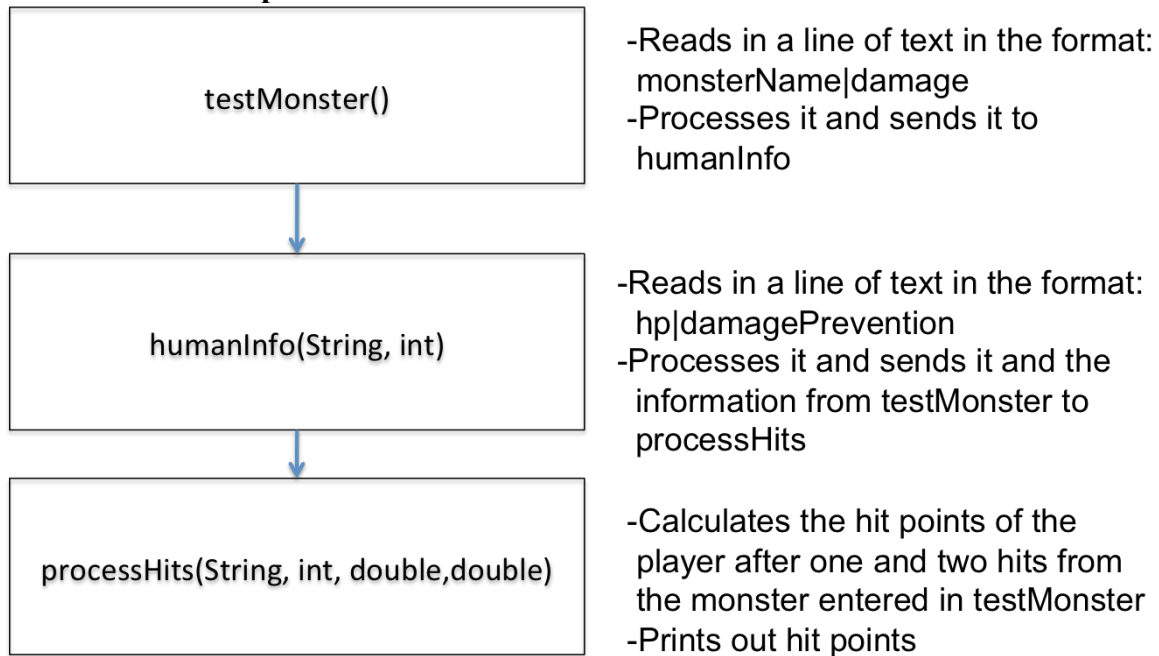
After seeing the value in your program, your bosses decide that they want a program that can be used to measure the HP of any player after one and two hits by any monster. In order to accomplish this, you have been given a driver program and an architecture diagram by a senior developer. The driver program prints out a header to indicate the program is starting, starts the processing for the monster hit test, and then prints out a footer indicating that the test is over. You must use the FunctionContainer file to implement the functions used in the driver class.

The architecture description lists the different functions used in the program, their parameters, and the flow of execution.

### Driver program:

```
public class Driver4Game
{
    public static void main(String[] args)
    {
        FunctionContainer program =
            new FunctionContainer();
        program.header();
        program.testMonster();
        program.footer();
    }
}
```

### Architecture description:



### Reiteration of steps required in the architecture:

You will need to write five functions: header, footer, testMonster, humanInfo, and processHits. The Driver4 program will call the header function, then the testMonster function. The testMonster function will call the humanInfo function, which will call the processHits function. Lastly, when Driver4 resumes executing code, it calls the footer method, then exits.

### Input format:

The input for the program is in the format of `value|value` and will be entered twice (in the command prompt). The first line represents the information for the monster. It will be in the format of `monsterName|damage` where the first part of the line is a string and the second part is an integer. The second line of input represents information about the player. It is in the format of `hitpoints|percentageDamageReductionFromArmor`. The first part is a double and the second is a double.

### Calculation:

In order to calculate the hit points for a player after a hit, you take the number of hit points the player has and subtract off the value of the damage from the hit minus the amount reduced from the original damage by the armor the player wears.

**Output Format:**

Your program must match the following format exactly. Obviously, you will need to replace the portions represented in < > with the appropriate information.

Monster testing program

Enter Monster Stats:

<input from command prompt>

Enter Player Stats:

<input from console>

Testing Monster named <from command prompt> that deals <from command prompt> damage per hit

After 1 hit the player has <calculated> hit points left

After 2 hits the player has <calculated> hit points left

End monster testing program

**Sample run:**

One possible input to the program is listed below. Make sure you test your program with more than just this one input case.

Monster testing program

Enter Monster Stats:

Orc|30

Enter Player Stats:

50|.5

Testing Monster named Orc that deals 30 damage per hit:

After 1 hit the player has 35.0 hit points left

After 2 hits the player has 20.0 hit points left

End monster testing program

*Hints:*

Write all of the functions with blank bodies (this will allow your program to run).

Write the header and footer functions first.

See me or a tutor if you need help.

After you are finished writing and testing your program, log into Desire 2 Learn and submit the java file under the homework 3 assignment. You can demo the homework the next school day after it is due, or it will be graded automatically. If you choose not to demo your homework you cannot contest the grade you receive.

**Grading rubric**

1 point – Compiles

1 point – Prints out header and footer properly

2 points – Produces correct output that conforms to the format described above.

1 point – Follows the design listed in the architecture description