

## Programming Assignments

For this assignment, use a SEPARATE Java file for each requirement (not sub-requirements)!

The main purpose of this assignment is to write classes to hold an Item, a Weapon, or a Food item. You will also write a Person class that interacts with these items.

Food and Weapon will inherit from Item; **do NOT replicate data in Food or Weapon that already exists in Item!** For example, Item already stores the name of the Item. Food should NOT have its own name variable; it should use the one from Item (either directly as a protected variable or indirectly through getter/setter methods).

For this assignment, **THE MAIN PROGRAM, SimpleAdventure.java, IS ALREADY WRITTEN FOR YOU.** It can be downloaded from Blackboard OR the course webpage. Your other classes/interfaces **MUST** work with this program.

Apart from the package name, **do NOT** modify SimpleAdventure.java!

Make sure your prompts and printouts match **EXACTLY** to the specification (**INCLUDING** capitalization, punctuation, spaces and newlines)!

Similarly, your java files / class names / method names **MUST** match the spelling and capitalization **EXACTLY!**

With the exception of constants, **NO DATA IN YOUR CLASS SHOULD BE PUBLIC OR DEFAULT/PACKAGE VISIBLE.** Use private or protected!

You can use the checkboxes to track whether you've met each requirement.

#	<b>Requirements</b>	
1	Create a class named <b>Person</b> with the following <b>public NON-STATIC</b> instance methods:	
	<b>public Person(String name)</b>	
	This CONSTRUCTOR takes the name of the Person and stores it in the class.	
	The health should also be set to a default value of 100 (int).	
	<b>public String getName()</b>	
	Returns the stored name.	
	<b>public int getHealth()</b>	

	Returns the stored health.	
	<b>public String toString()</b>	
	Returns a String with the following format: if the name were "Bob" and the health were 89, then the String should contain: "Name: Bob\nHealth: 89\n" Note the newlines after each line!	
	<b>This does NOT print anything! This ONLY returns a String!</b>	
	<b>public boolean isAlive()</b>	
	Returns true if health is NOT zero.	
	<b>public boolean heal(int boost)</b>	
	If the Person is still alive, add the boost to the health, making sure that the health does not exceed 100, and then return true.	
	Otherwise, if the Person is NOT alive, return false.	
	<b>public boolean defends(int damage)</b>	
	Subtract damage from health, making sure the health does not drop below zero.	
	Return whether the Person is alive.	

2	Create a class named <b>Item</b> with the following public <b>NON-STATIC</b> instance methods:	
	<b>public Item(String name, double weight)</b>	
	This CONSTRUCTOR takes the name and weight of the item and stores it in the class.	
	<b>public String getName()</b>	
	Returns the stored name.	
	<b>public double getWeight()</b>	
	Returns the stored weight.	
	<b>public void setName(String name)</b>	
	Sets the stored name.	
	<b>public void setWeight(double weight)</b>	
	Sets the stored weight.	
	<b>public String toString()</b>	

	Returns a String with the following format: if the name were "Fork" and the weight were 0.45, then the String should contain: "Name: Fork\nWeight: 0.45\n" Note the newlines after each line!	
	<b>This does NOT print anything! This ONLY returns a String!</b>	
	<b>public boolean use(Object target)</b>	
	Prints "Not usable"	
	Returns false	
3	Create a class named <b>Food</b> <i>that inherits from Item</i> with the following public <b>NON-STATIC</b> instance methods:	
	<b>public Food(String name, double weight, int health)</b>	
	This CONSTRUCTOR takes the name, weight, and health of the item and stores it in the class.	
	Again, use the name and weight data from the parent class <b>Item</b> (directly or indirectly)!	
	<b>public int getHealth()</b>	
	Returns the stored health.	
	<b>public void setHealth(int health)</b>	
	Sets the stored health.	

	<b>public String toString()</b>	
	Returns a String with the following format: if the name were "Eggs", the weight were 1.7, and the health were 45, then the String should contain: "Name: Eggs\nWeight: 1.7\nHealth: 45\n" Note the newlines after each line!	
	<b>This does NOT print anything! This ONLY returns a String!</b>	
	<b>public boolean use(Object target)</b>	
	If the target is NOT an instance of Person, return false.	
	Otherwise, call heal() on the Person with the health of this Food item. Let's assume the Person reference is p.	
	If heal() returned true, print p.getName() + " ate " + getName() + " for " + health + " health!" and return true	
	If heal() returned false, print p.getName() + " cannot be healed!" and return false	

4	<b>Create a class named <i>Weapon</i> that inherits from <i>Item</i> with the following public NON-STATIC instance methods:</b>	
	<b>public Weapon(String name, double weight, int damage)</b>	
	This CONSTRUCTOR takes the name, weight, and damage of the item and stores it in the class.	
	Again, use the name and weight data from the parent class <i>Item</i> (directly or indirectly)!	
	<b>public int getDamage()</b>	
	Returns the stored damage.	
	<b>public void setDamage(int damage)</b>	
	Sets the stored damage.	
	<b>public String toString()</b>	
	Returns a String with the following format: if the name were "BFG", the weight were 28.1, and the damage were 9000, then the String should contain: "Name: BFG\nWeight: 28.1\nDamage: 9000\n" Note the newlines after each line!	
	<b>This does NOT print anything! This ONLY returns a String!</b>	

	<b>public boolean use(Object target)</b>	
	If the target is NOT an instance of Person, return false.	
	Otherwise, let's assume the Person reference is p. Print "Attack " + p.getName() + " with " + getName() + " for " + damage + " damage!"	
	Call defends() on the Person with the damage of this Weapon item.	
	If defends() returned true, print p.getName() + " lives!"	
	If defends() returned false, p.getName() + " is dead!"	
	Return true	
<b>5</b>	<b>The main program, SimpleAdventure.java, is already provided for you.</b>	
	The ONLY thing you should modify with this code is the package name. Beyond that, <b>your code MUST run with this program.</b>	
	See section "Program Output of SimpleAdventure" for EXACTLY what this program should output.	

## Program Output of SimpleAdventure

The health of Frodo Baggins is 100  
Frodo Baggins is alive.  
Name: Frodo Baggins  
Health: 100

The health of Samwise Gamgee is 100  
Samwise Gamgee is alive.  
Name: Samwise Gamgee  
Health: 100

The health of Smeagol is 100  
Smeagol is alive.  
Name: Smeagol  
Health: 100

The item Sting has weight 1.5  
It also does 30 damage  
Name: Sting  
Weight: 1.5  
Damage: 30

The item Taters has weight 0.3  
It also heals 45  
Name: Taters  
Weight: 0.3  
Health: 45

The item There and Back Again has weight 5.3  
Name: There and Back Again  
Weight: 5.3

The item Glowing Sting has weight 1.4  
It also does 60 damage  
Name: Glowing Sting  
Weight: 1.4  
Damage: 60

The item <sup>162</sup>POTATOES has weight 0.2  
It also heals 56  
Name: POTATOES  
Weight: 0.2  
Health: 56

The item There and Back Again: And What Happened After has weight 15.7  
Name: There and Back Again: And What Happened After  
Weight: 15.7

Use book on sword...  
Not usable  
Can use? false

Use book on frodo...  
Not usable  
Can use? false

Use book on null...  
Not usable  
Can use? false

Use sword on gollum...  
Attack Smeagol with Glowing Sting for 60 damage!  
Smeagol lives!  
Can use? true

The health of Smeagol is 40  
Smeagol is alive.  
Name: Smeagol  
Health: 40

Use sword on book...  
Can use? false

Use sword on null...  
Can use? false

Use taters on null...  
Can use? false

Use taters on book...  
Can use? false

Use taters on gollums...  
Smeagol ate POTATOES for 56 health!  
Can use? true



The health of Smeagol is 96  
Smeagol is alive.  
Name: Smeagol  
Health: 96

Use sword on gollum...  
Attack Smeagol with Glowing Sting for 60 damage!  
Smeagol lives!  
The health of Smeagol is 36  
Smeagol is alive.  
Name: Smeagol  
Health: 36

Use sword on gollum again...  
Attack Smeagol with Glowing Sting for 60 damage!  
Smeagol is dead!  
The health of Smeagol is 0  
Smeagol has passed on.  
Name: Smeagol  
Health: 0

Use taters on gollum...  
Smeagol cannot be healed!  
Can use? false

NOTE: ONLY SimpleAdventure has a main() method!