CS 249: Assignment 6

Inheritance and Polymorphism

Theory Questions (24%)

Note: for any UML diagrams, you can use a UML drawing utility like UMLet.
For ALL classes coded and designed, make sure they are:

- Clear, consistent, and have central purpose
- Maintaining a good abstraction
- Enforcing encapsulation

1. (2%) Given a superclass "Animal" and a subclass "Walrus", write the CODE for the class definition of "Walrus". It should contain NO data or methods; just show the class definition such that "Walrus" inherits from "Animal".

   ```
   class Walrus extends Animal
   ```

2. (2%) Given the code below, what is this an example of?

   (a) Upcasting
   (b) Downcasting

   (a) Upcasting

   ```
   Animal a = new Walrus();
   ```

3. (2%) Is the code in the previous question legal? Briefly explain your answer.

   Upcasting always executes so this is perfectly legal

4. (2%) **Design and draw the UML diagrams for the following classes: Also draw the inheritance relationship!**

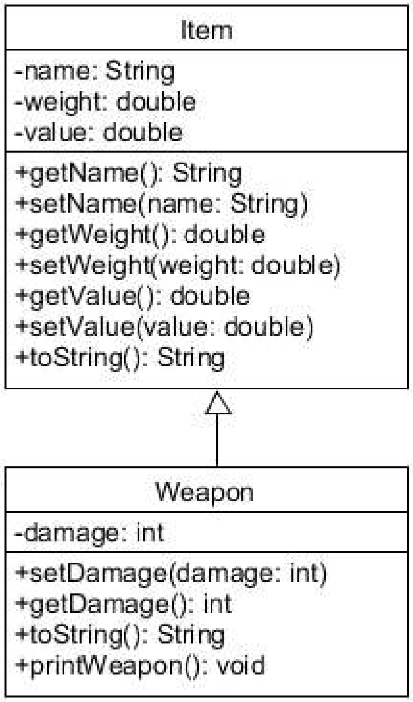
Name: **Item**
It should contain the following data:

- Name
- Weight
- Value

It should have the following functionality:

- No-arg constructor (name = "", weight = 0, value = 0)
- Constructor that takes name, weight, and value
- Getter/setter functions for data
- Override toString() to return a String with the following format:
  Example: if name = "Sword", weight = 5, and value = 100:

  Name: Sword
  Weight: 5
  Value: 100

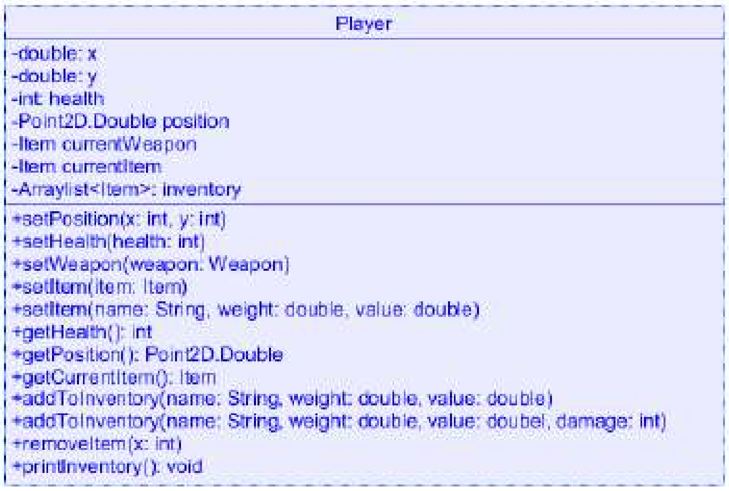


Name: **Weapon**
Use the same class from the last assignment, BUT make the following modifications:

- Inherit from Item
- Remove name field
- Add no-args constructor (damage = 0)
- Change existing constructor to take (name, weight, value, damage)
- Call the appropriate super() constructor where necessary
- Remove getter/setter functions for name (already in Item)
- Override toString() to return super.toString() + the damage:
  Example: if name = "Sword", weight = 5, value = 100, and damage = 30:

  Name: Sword
  Weight: 5
  Value: 100
  Damage: 30

5. (2%) **Modify the Player class and draw the new UML diagram for it (you do not need to draw any relationships here):**



```
Player
-double: x
-double: y
-int: health
-Point2D.Double position
-Item currentWeapon
-Item currentItem
-Arraylist<Item>: inventory
+setPosition(x: int, y: int)
+setHealth(health: int)
+setWeapon(weapon: Weapon)
+setItem(item: Item)
+setItem(name: String, weight: double, value: double)
+getHealth(): int
+getPosition(): Point2D.Double
+getCurrentItem(): Item
+addToInventory(name: String, weight: double, value: double)
+addToInventory(name: String, weight: double, value: doubel, damage: int)
+removeItem(x: int)
+printInventory(): void
```

6. (2%) A child class inherits private methods from the superclass.

   (a) True   (b) False

   (b) False

7. (2%) In Java, a subclass may only extend ONE superclass.

   (a) True   (a) True

   (b) False

8. (2%) A protected field in the superclass is accessible by a subclass, EVEN if the subclass is in a DIFFERENT package than the superclass.

   (a) True   (a) True

   (b) False

9. (2%) What class is the ancestor of ALL other Java classes? (I.e., it is at the very top of the inheritance tree?)

   Object class

10. (2%) What Java operator allows me to check what class a given object is?

instanceOf

11. (2%) When is the no-args **super()** constructor ***implicitly*** called?

   (a) Never; it is only called when the programmer explicitly states it.
   (b) Only if the current constructor does not explicitly make a call to another constructor or superclass constructor. (b)
   (c) Under all circumstances.

12. (2%) To **override** a method, what must be the same?

   (a) The signature (b) The return type
   (b) The return type
   (c) Both (a) and (b)