

CSE102 Assignment 1

Game Characters and Equipment System

Due: Sunday, 16.03.2025 by 23:59

Deliverables

Submit a single Java file named `Assignment01_{StudentNumber}.java` via Moodle by the due date and time specified. Late submissions will be subject to the course's late policy.

Overview

In this assignment, you will develop a game system that models three types of characters—Warrior, Mage, and Archer—each with its own inventory and equipment management system. Your program should demonstrate the basic Object-Oriented Programming (OOP) principles of **inheritance** and **encapsulation**. Each derived class defines its own uniquely named methods (e.g., `warriorAttack`, `mageAttack`, `archerAttack`) and inherits common inventory/equipment methods from the base `Character` class.

Specifications

Your implementation must include the following classes and functionality:

1. Item Class

- **Attributes:**

- `name: String` — The item's name (e.g., "Sword of Might").
- `bonusType: String` — The type of bonus provided. Valid values are: "HEALTH", "MANA", "ACCURACY", "ATTACK".
- `bonusValue: int` — The bonus value (e.g., 5).

- **Methods:**

- `Item(String name, String bonusType, int bonusValue):` Initializes the attributes.
- **Getter methods:** `getName()`, `getBonusType()`, `getBonusValue()`.
- `toString():` Returns a string such as: "Sword of Might [ATTACK Bonus: 5]".

2. Character Class (Base Class)

- **Attributes:**

- `name: String` — The character's name (e.g., "Conan").
- `health: int` — The current health points.
- `attackPower: int` — The base attack power.

- `inventory` and `equipment`: These fields (arrays of `Item`) are declared as protected along with their capacities (e.g., `inventoryCapacity`, `equipmentCapacity`). Derived classes will initialize these arrays with sizes specific to each character type.

- **Methods:**

- `Character(String name, int health, int attackPower)`: Initializes the attributes.
- `takeDamage(int damage)`: Reduces health by the specified damage (ensuring health remains non-negative) and prints the current health as “Current Health: X.”
- `toString()`: Returns a summary such as “Conan [Current Health: 85, Base Attack: 15].”

- **Inventory/Equipment Methods:**

- * `addItemToInventory(Item item)`: Adds an item to the first available slot.
- * `removeItemFromInventory(int index)`: Removes an item from a specified index and shifts the array.
- * `equipItemFromInventory(int invIndex)`: Transfers an item from inventory to an available equipment slot.
Example output: “Conan equips Sword of Might from inventory. (Bonus ATTACK: 5)”.
- * `unequipItem(int equipIndex)`: Moves an item from equipment back to inventory.
Example output: “Merlin removes Mystic Staff from equipment. (Bonus MANA: 0)”.

3. Derived Classes: Warrior, Mage, and Archer

Note: Each class defines its own uniquely named methods.

a) Warrior Class

- **Attributes:**

- `armor`: `double` — Reduces incoming damage.
- `inventory`: `Item[]` — Array with capacity 6.
- `equipment`: `Item[]` — Array with 3 slots.

- **Methods:**

- `warriorTakeDamage(int damage, boolean ignoreArmor)`:
 1. If `ignoreArmor` is true or `armor` is 0, call `takeDamage(damage)`.
 2. Otherwise, calculate damage reduction (10% if armor value is more or equal to 50; 5% otherwise) and apply the reduced damage.
 3. Reduce `armor` by 10 and print “Current Armor: X.”
 4. Print “Current Health: X” after damage is applied.
- `warriorAttack(Character target)`:
 1. Compute total attack power = base attack + any ATTACK bonus from equipment.
 2. Attack the target using `takeDamage(totalAttackPower)`.

b) Mage Class**• Attributes:**

- `mana: int` — Mana available for casting spells.
- `inventory: Item[]` — Array with capacity 2.
- `equipment: Item[]` — Array with 1 slot.

• Method:

- `mageAttack(Character target):`
 1. If mana value is greater or equal to 10, compute spell damage = total attack power (base + ATTACK bonus) + 5.
 2. Apply damage to the target and reduce mana by 10.
 3. Print “Mage’s Remaining Mana: X” (showing current mana).
 4. Otherwise, perform a basic attack.
- Inventory and equipment methods work similarly to those in the Warrior class.

c) Archer Class**• Attributes:**

- `accuracy: int` — Determines the chance for a critical hit.
- `inventory: Item[]` — Array with capacity 4.
- `equipment: Item[]` — Array with 2 slots.

• Method:

- `archerAttack(Character target):`
 1. Calculate total attack power = base attack + any ATTACK bonus.
 2. If (base accuracy + any ACCURACY bonus) is greater than 80, add an extra 5 damage.
 3. Attack the target with the computed damage.
- Inventory and equipment methods are implemented similarly to those in the Warrior class.

Design, Code, and Test

Design: Your program does not require a `main()` method. You are only responsible for creating the classes described above (Item, Character, Warrior, Mage, and Archer) in a single Java file named `Assignment01_{StudentNumber}.java`. An example of how your program should operate is provided below.

Code: All Java classes must be included in one file. Your code should be clearly organized with header comments that explain the purpose of each class and method. Use Java’s standard libraries (e.g., `Math` for rounding) if needed; include a comment indicating any such usage.

Test: You are responsible for testing your program. Although you should not submit a `main()` method in the final file, you must create your own tests. For example, you can simulate a short combat scenario where:

- a) The Warrior attacks the Mage using `warriorAttack()`.
- b) The Mage attacks the Archer using `mageAttack()`.

c) The Archer attacks the Warrior using `archerAttack()`.

After each action, only the current values (e.g., “Current Health: 50”, “Current Armor: 40”, “Mage’s Remaining Mana: 20”) should be printed. Also, ensure that inventory and equipment management operations (addition, removal, equipping, unequipping) display concise messages indicating the affected bonus (e.g., “Conan equips Sword of Might from inventory. (Bonus ATTACK: 5)”).

```
class TestGame {
    Run | Debug
    public static void main(String[] args) {
        // Create characters.
        Warrior warrior = new Warrior(name:"Conan", health:100, attackPower:15, armor:60);
        Mage mage = new Mage(name:"Merlin", health:80, attackPower:10, mana:30);
        Archer archer = new Archer(name:"Robin", health:70, attackPower:12, accuracy:75);

        // Create and equip items.
        warrior.addItemToInventory(new Item(name:"Sword of Might", bonusType:"ATTACK", bonusValue:5));
        warrior.addItemToInventory(new Item(name:"Shield of Valor", bonusType:"HEALTH", bonusValue:15));
        warrior.addItemToInventory(new Item(name:"Rune of Power", bonusType:"ACCURACY", bonusValue:5));
        warrior.equipItemFromInventory(invIndex:0); // Equips Sword of Might (Bonus ATTACK: 5)
        warrior.equipItemFromInventory(invIndex:1); // Equips Shield of Valor (Bonus HEALTH: 15)

        mage.addItemToInventory(new Item(name:"Mystic Staff", bonusType:"MANA", bonusValue:10));
        mage.addItemToInventory(new Item(name:"Rune of Fury", bonusType:"ATTACK", bonusValue:5));
        mage.equipItemFromInventory(invIndex:0); // Equips Mystic Staff (Bonus MANA: 10)

        archer.addItemToInventory(new Item(name:"Longbow", bonusType:"ATTACK", bonusValue:4));
        archer.addItemToInventory(new Item(name:"Quiver of Precision", bonusType:"ACCURACY", bonusValue:5));
        archer.equipItemFromInventory(invIndex:0); // Equips Longbow (Bonus ATTACK: 4)
        archer.equipItemFromInventory(invIndex:1); // Equips Quiver of Precision (Bonus ACCURACY: 5)

        // Short combat simulation.
        System.out.println(x:"\n--- Combat Simulation ---");
        // Warrior attacks Mage.
        warrior.warriorAttack(mage);
        // Mage attacks Archer.
        mage.mageAttack(archer);
        // Archer attacks Warrior.
        archer.archerAttack(warrior);

        // Removal test: Mage unequips its equipment.
        System.out.println(x:"\n--- Removal Test ---");
        mage.unequipItem(equipIndex:0); // Should remove Mystic Staff and show bonus as 0.

        // Final status.
        System.out.println(x:"\n-- Final Stats --");
        System.out.println(warrior);
        System.out.println(mage);
        System.out.println(archer);
    }
}
```

Test Case Example

```
Conan equips Sword of Might from inventory. (Bonus ATTACK: 5)
Conan equips Shield of Valor from inventory. (Bonus HEALTH: 15)
Merlin equips Mystic Staff from inventory. (Bonus MANA: 10)
Robin equips Longbow from inventory. (Bonus ATTACK: 4)
Robin equips Quiver of Precision from inventory. (Bonus ACCURACY: 5)

--- Combat Simulation ---
Conan attacks Merlin with total power 20
Merlin takes 20 damage. (Current Health: 60)
Merlin casts a spell on Robin dealing 15 damage!
Robin takes 15 damage. (Current Health: 55)
Merlin's Remaining Mana: 20
Robin attacks Conan with total power 21
Conan damage reduced.
Conan takes 19 damage. (Current Health: 81)
Conan's Remaining Armor: 50.0

--- Removal Test ---
Merlin removes Mystic Staff from equipment. (Bonus MANA: 0)

-- Final Stats --
Warrior Conan [Current Health: 81, Total Attack: 20, Current Armor: 50.0, Bonus HEALTH: 15]
Mage Merlin [Current Health: 60, Total Attack: 10, Current Mana: 20, Bonus MANA: 0]
Archer Robin [Current Health: 55, Total Attack: 16, Accuracy: 75, Bonus ACCURACY: 5]
```

Output Example