

Computer Programming Lab, *Spring 2021*
Empire Building: **Quiz 1**

Tuesday 11/5/2021 5th

Decryption Password

UN40@*AS

In this quiz, we will modify the structure of the game to set the foundation for the following potential change.

1 Naming and privacy conventions

Please note that all your class attributes must be `private` and all methods should be `public` unless otherwise stated. You should implement the appropriate setters and getters conforming with the access constraints. Throughout the whole milestone, if a variable is said to be READ then we are allowed to get its value. If the variable is said to be WRITE then we are allowed to change its value. Please note that getters and setters should match the Java naming conventions. If the instance variable is of type boolean, the getter method name starts by `is` followed by the **exact** name of the instance variable. Otherwise, the method name starts with the verb (get or set) followed by the **exact** name of the instance variable; the first letter of the instance variable should be capitalized. Please note that the method names are case-sensitive.

For your task you need to implement the following:

2 PassiveUnit

You need to add the following Class to your code:

Name : `PassiveUnit`

Package : `units`

Type : Class

Description : A Class representing the educational buildings. No object of `PassiveUnit` can be instantiated.

2.1 Class attributes

All the class attributes are READ and WRITE unless otherwise specified.

1. `int level`: The level of the passive unit.
2. `double foodConsumption`: The food consumption that this unit needs each turn

2.2 Constructors

1. `public PassiveUnit(int level, double foodConsumption)`: Constructor that initializes the attributes of a PassiveUnit object.

3 Farmer

You need to add the following Class to your code:

Name : `Farmer`

Package : `units`

Type : Class

Description : A subclass of PassiveUnit representing a farmer.

3.1 Class attributes

All the class attributes are READ and WRITE unless otherwise specified.

1. `double foodProduction`: The food amount that this farmer produces.
2. `int cooldown`: the cooldown in number of turns for this farmer to produce more food.

3.2 Constructors

1. `public Farmer(int level, double foodConsumption, double foodProduction, int cooldown)`: Constructor that initializes the attributes of a Farmer object.