

MOCK ASSESSMENT 3 - OOP, Abstract Classes

Core concepts: Classes, Class methods

OVERVIEW

Create a C# console application that utilizes OOP concepts..
Your completed application should include classes and methods that perform the required actions and follow proper naming conventions.

BUILD SPECIFICATIONS

Pay special attention to the items in bold. **You must use these in your program exactly, including capitalization**, in order to get the points.

For this challenge, you will need to create a new C# Project named **MockAssessment3**. All classes must be created in the **namespace MockAssessment3**.

In this project you will create **Villager** and **Town** classes to see if the Villagers can produce enough food to make it through the winter. **Villager** will be abstract and will have 2 children: **Slacker** and **Farmer**.

- **class Town**
 - Properties
 - **public List<Villager> Villagers**
 - Constructor
 - Add 1 Farmer and 3 Slackers to the Villagers List
 - Methods
 - **public Harvest()** - returns an **int**
 - Find the total of all the Villagers' **Farm()** methods
 - Return that total
 - **public CalcFoodConsumption()** - returns an **int**
 - Find the total of all the Villagers' **Hunger** properties
 - Return that total
 - **public SurviveTheWinter()** - returns a **bool**
 - Call and store the result of the **Harvest()** method
 - Call and store the result of the **CalcFoodConsumption()** method
 - Return **true** if the value from CalcFoodConsumption is less than or equal to Harvest.
 - Return **false** if the value from CalcFoodConsumption is greater than Harvest.
- **abstract class Villager**
 - Properties

- **public Int Hunger**
 - Methods
 - **abstract Farm()** - returns an **int**
- **class Farmer** which will be a child of **Villager**
 - Constructor
 - Set **Hunger** to **1**
 - Methods
 - Override **Farm()** return **2**
- **class Slacker** which will be a child of **Villager**
 - Constructor
 - Set **Hunger** to **3**
 - Methods
 - Override **Farm()** return **0**