Greg's OOPG question (Console Visual C#)

Background

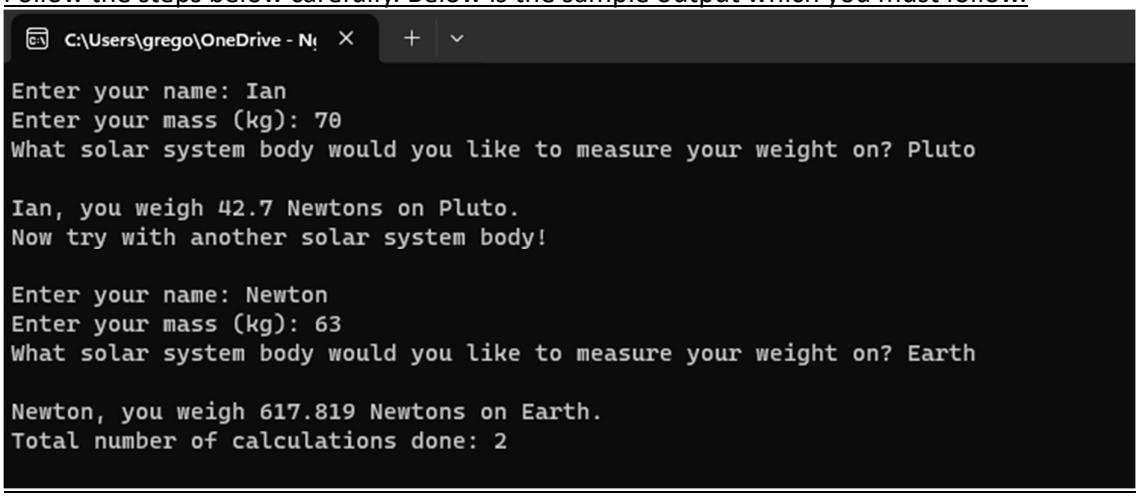
Many use the terms 'weight' and 'mass' interchangeably. Both actually mean different things. Mass is the amount of matter in an object, while weight is the amount of force a body exerts due to gravitational pull. Therefore, mass and weight have different units. The SI unit for mass is kilogrammes (kg) and the SI unit for weight is in Newtons (N) since it represents a quantity of a force.

$$w = mg$$

Where:
w = your weight in Newtons (N)
m = your mass in kg
g = the body's gravitational constant, in N/kg.

The program that you will be writing will give you two chances to calculate your weight on some of the eight planets, plus the Moon.

Follow the steps below carefully. Below is the sample output which you must follow.



a. Write a class `SolarSystemCalc` with the following members: a private string `strObjectName`, a protected float `fltGravityConst`, and a protected static int `intCount`, and a private float `fltMass`.
b. Write a read-only property `Count` that returns `intCount`.
c. Write a read-only property `ObjectName` that returns `strObjectName`.
d. Write a constructor that takes in two parameter and initialises `strObjectName`, as well as `fltMass`. The constructor will be used to set the value `fltGravityConst`, based on the value of `strObjectName`.

Body (`strObjectName`)	Value of g (`fltGravityConst`)
Earth	9.80665
Moon	1.62500
Mars	3.72026
Saturn	11.19000
Pluto	0.61000

e. Write a method `CalcWeight()` to calculate the weight of the person on the solar system object based on the equation above. This means the return value is a float.
f. Derive a class `MyInfo` from `SolarSystemCalc`. It has the following members: a private string `strName`.
g. Write a read-only property `Name` that returns the value of `strName`.
h. Write a constructor to initialise `strName`, and of course remember to reference the base constructor. Additionally, the constructor should increment `intCount` by 1. That means the constructor takes in three parameters; one for the derived class, and two for the base class.
i. In `class Program > main()`, get user to input their name, mass and the solar system body they would like to measure their weight on.
j. Initialise and instantiate `objP1` of class `MyInfo`.
k. Display the weight of the person on the solar system body.
l. Again, get user to input their name, mass and the solar system body they would like to measure their weight on.
m. Initialise and instantiate `objP2` of class `MyInfo`.
n. Display the weight of the person on the solar system body by calling `CalcWeight()`
o. Display the total number of calculations done.