Final project housing app

**Final Project – Housing Application**

You are creating a housing application for a property manager. This program design includes the following:

A base class named Housing:

Include data characteristics such as address, type of construction and year built.

Contains a virtual method named ProjectedRentalAmt, returns decimal. Create this method in the base class to return 0.

Contains an override ToString method that prints all of the relevant class information as a string.

Define an interface named IUnits:

Contains a method that returns the number of units, as int.

A class named MultiUnit, inherits from Housing, Implements the IUnits interface.

Include data characteristics such as complex name, number of units, and rent amount per unit

Contains an override method ProjectedRentalAmt(), returns a decimal. Take the rent amount per unit multiplied by the number of units, multiplied by 12 months.

Contains the method GetNumUnits, returns int – required by the Interface

Contains an override ToString method that prints all of the relevant class information as a string.

Class SingleFamily, inherits from Housing

Single- Family should include characteristics such as the rent amount, size in square feet, number of bedrooms, number of bathrooms, whether there is a porch, and availability of garage.

Contains an override method ProjectedRentalAmt(), returns a decimal. Take the rent amount multiplied by 12 months.

Contains an override ToString method that prints all of the relevant class information as a string.

Each class should contain a constructor that takes all properties and sets up the values. Use monetary formatting where appropriate.

**Tests**

Once you have the class and interface structure completed, write the following tests to show your code working.

Single Family Test

Create a test method that creates 5 instances of the SingleFamily class using the constructor. Put these into a List<SingleFamily>.

Create a loop that reads through each of these houses and prints the ToString information

MultiUnits Test

Create a test method that creates 5 instances of the MultiUnits class using the constructor. Put these into a List< MultiUnits >.

Create a loop that reads through each of these objects and prints the ToString information.

Combined Test

Take the 5 SingleFamily objects and 5 MultiUnits objects you created in the previous tests and use them again in a third test method.  Create a List<Housing> to hold all 10 objects.

Create a loop that reads through each of these and prints the addresses and projected rental amount.