CIS 400 Practice Exam I

Name: _____ WID: _____ Section: _____

Vocabulary

The following questions deal with object-orientation on the theoretical level, not a specific language

Q1. What is encapsulation?

Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Q2. Describe the relationship between classes and objects:

Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Q3. Describe the purpose of Inheritance:

Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

C# Keywords

The following questions deal specifically with the C# language

Q4. What does the keyword “protected” mean when used with a field?

Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Q5. What does the “abstract” keyword mean when used with a class?

Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Q6. What does the “new” keyword mean when used with a method?

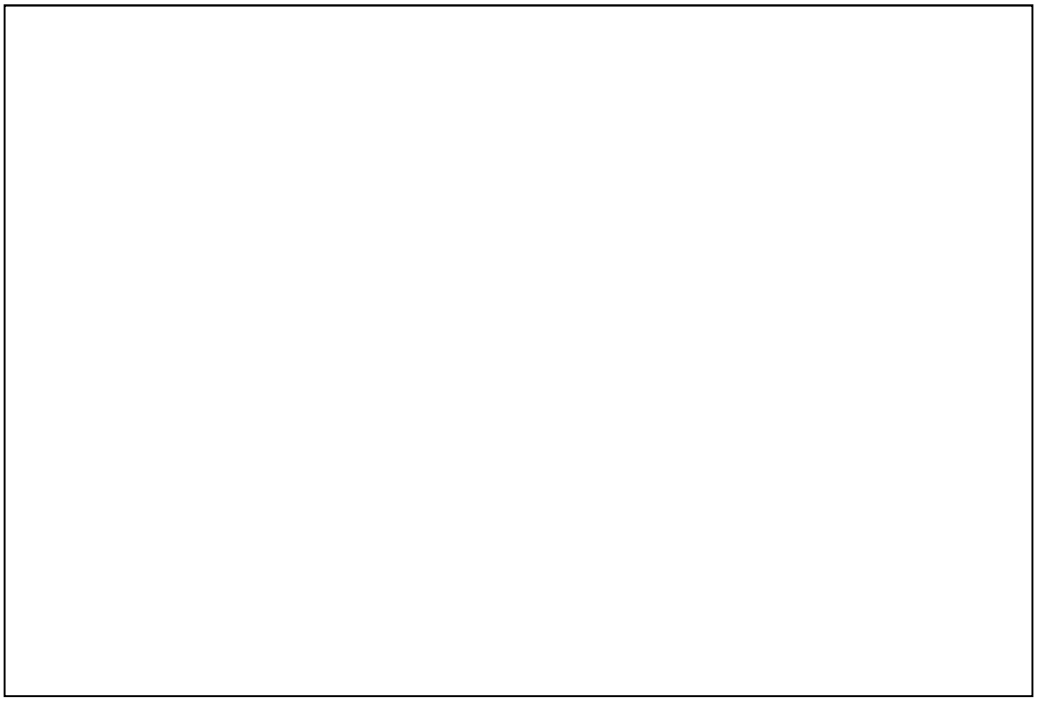
Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Coding Questions

The following questions deal with this code:

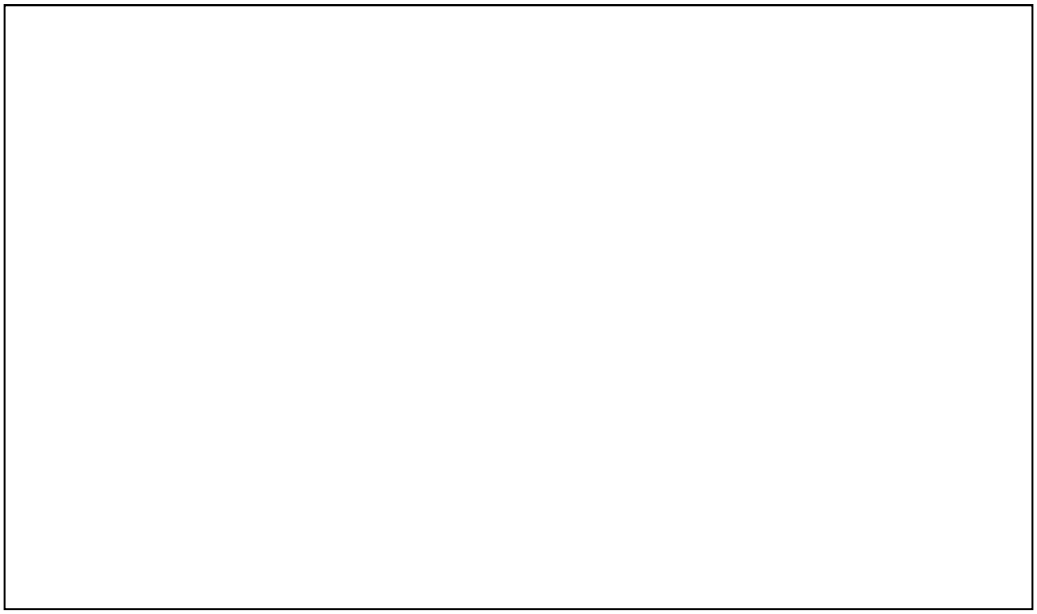
```
    /// <summary>
    /// A class representing a "smart" room equipped with a communicating thermostat.
    /// </summary>
    public class SmartRoom
    {
        // Private backing variables
        private Thermostat thermostat;
        private double squareFootage = 0;
        /// <summary>
        /// Gets or sets the square footage, which must be a positive number.
        /// </summary>
        public double SquareFootage
        {
            get { return squareFootage; }
            set
            {
                if (value > 0) squareFootage = value;
            }
        }
        /// <summary>
        /// Gets the room's current temperature
        /// </summary>
        public double CurrentTemperature
        {
            get { return thermostat.CurrentTemperature }
        }
        /// <summary>
        /// Gets and sets the room's target temperature
        /// </summary>
        public double TargetTemperature
        {
            get { return thermostat.TargetTemperature; }
            set { thermostat.TargetTemperature = value; }
        }
        /// <summary>
        /// Constructs a new Room of specified size equipped with provided thermostat.
        /// </Summary>
        /// <param name="squareFootage">
        /// The square footage of the room, as a positive double.
        /// </param>
        /// <param name="thermostat">
        /// An interface for communicating with the room's thermostat.
        /// </param>
        public SmartRoom(double squareFootage, Thermostat thermostat)
        {
            if (squareFootage < 0)
                throw new ArgumentException("Square footage must be positive.");
            if (thermostat == null)
                throw new ArgumentNullException("A thermostat must be provided.");
            this.squareFootage = squareFootage;
            this.thermostat = thermostat;
        }
    }
}
```

Q7. Draw the UML diagram for the provided SmartRoom class:



Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Q8. Write a new read-only accessor, DeltaT, which provides the difference between the current and target temperature for the room.



Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Q9. Complete the supplied test method to assert that attempting to set a negative square footage will not change the SmartRoom's state:

```
using Xunit;

namespace SmartHome
{
    public class SmartRoomTest
    {
        [Fact]
        public void ThermostatShouldNotAcceptNegativeNumbers()
        {

        }
    }
}
```

Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all

Q10. Create a derived class from SmartRoom named LitSmartRoom that includes four Boolean properties (LightA, LightB, LightC, and LightD) with getters and setters, plus the methods SwitchAllLightsOn() and SwitchAllLightsOff() that set all four to true and false, respectively.

Confidence (circle one): very confident - (5) (4) (3) (2) (1) - Not confident at all