```
/// Written by Camille Rasmussen
/// UID: u0717763
/// CS 3500 Fall 2014

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using SpreadsheetUtilities;

namespace SS
{
    /// <summary>
    /// The Cell class provides an object place holder in the spreadsheet class.
    /// Each cell is comprised of a name, contents, and value.
    ///
    /// The value of a cell can be (1) a string, (2) a double, or (3) a FormulaError.
    /// (By analogy, the value of an Excel cell is what is displayed in that cell's position
    /// in the grid.)
    ///
    /// If a cell's contents is a string, its value is that string.
    ///
    /// If a cell's contents is a double, its value is that double.
    ///
    /// If a cell's contents is a Formula, its value is either a double or a FormulaError,
    /// as reported by the Evaluate method of the Formula class.  The value of a Formula,
    /// of course, can depend on the values of variables.  The value of a variable is the
    /// value of the spreadsheet cell it names (if that cell's value is a double) or
    /// is undefined (otherwise).
    /// </summary>
    21 references
    public class Cell
    {
        // the name of the cell
        private string name;
        // the contents vs value ex. formula vs. evaluated formula
        private object contents, value;

        /// <summary>
        /// The Cell constructor for a double
        /// Sets name, contents, and value appropriately
        /// </summary>
        /// <param name="name"></param>
        /// <param name="number"></param>
        1 reference
        public Cell(string name, double number)
        {
            this.name = name;
            this.contents = number;
            this.value = number;
        }

        /// <summary>
        /// The Cell constructor for a text string
        /// Sets name, contents, and value appropriately
        /// </summary>
        /// <param name="name"></param>
        /// <param name="text"></param>
        1 reference
        public Cell(string name, string text)
        {
            this.name = name;
            this.contents = text;
            this.value = text;
        }

        /// <summary>
        /// The Cell constructor for a Formula
        /// Formula gets passed in omitting the = at the beginning
        /// Sets name and contents appropriately, and value is
        /// calculated using the Formula.Evaluate() method passing
        /// in the LookUp function
        /// </summary>
        /// <param name="name"></param>
        /// <param name="formula"></param>
        /// <param name="lookup"></param>
        1 reference
        public Cell(string name, Formula formula, Func<string, double> lookup)
        {
            this.name = name;
            this.contents = formula;
            this.value = formula.Evaluate(lookup);
        }

        /// <summary>
        /// Gets and returns the name of the cell
        /// </summary>
        /// <returns></returns>
        9 references
        public string GetName()
        {
            return name;
        }

        /// <summary>
        /// Gets and returns the contents of the cell
        /// </summary>
        /// <returns></returns>
        10 references
        public object GetContents()
        {
            return contents;
        }

        /// <summary>
        /// Sets the contents of a cell to this double
        /// </summary>
        /// <param name="number"></param>
        1 reference
        public void SetDoubleCellContents(double number)
        {
            this.contents = number;
            this.value = number;
        }

        /// <summary>
        /// Sets the contents of a cell to this text string
        /// </summary>
        /// <param name="text"></param>
        1 reference
        public void SetTextCellContents(string text)
        {
            this.contents = text;
            this.value = text;
        }

        /// <summary>
        /// Sets contents of a cell to this formula
        /// </summary>
        /// <param name="formula"></param>
        /// <param name="lookup"></param>
        1 reference
        public void SetFormulaCellContents(Formula formula, Func<string, double> lookup)
        {
            this.contents = formula;
        }

        /// <summary>
        /// Gets and returns this cells value
        /// </summary>
        /// <returns></returns>
        2 references
        public object GetValue()
```