```
//*********************************************************************************
//Program:     Methods_Documentation.cs
//Description: Calculates body mass index of user
//Date:        Oct. 21/2015
//Author:      JD Silver
//Course:      CMPE1300
//Class:       CNTA01
//*********************************************************************************

using System;
namespace Methods_Documentation
{
    class Program
    {
        static void Main(string[] args)
        {
            int feet = 0;           // user's height feet
            int inches = 0;         // user's height inches
            double weight = 0.0;    // user's weight in pounds
            double bmi = 0.0;       // calculated body mass index

            // input the height feet component
            feet = GetInt("Enter number of feet in your height: ", 0, 10);

            // input the height inches component
            inches = GetInt("Enter number of inches in your height: ", 0, 11);

            // input the weight in pounds
            weight = GetDouble("Enter your weight in pounds: ", 0.0, 500.0);

            // calculate the body mass index
            bmi = CalculateBMI(feet, inches, weight);

            // display bmi
            Console.WriteLine("BMI = {0:F1}", bmi);

            Console.ReadKey();
        }
```

Global Variables are not allowed except for GDIDrawer

Local variables documented using inline comments

Method calls commented above each call.

```
//********************************************************************************************
//Method:    static private int GetInt(string prompt, int min, int max)
//Purpose:   Inputs an integer value with error and range checking
//Parameters:string prompt - prompt to display to the user
//           int min - minimum value accepted
//           int max - maximum value accepted
//Returns:   int - value accepted by the method
//*********************************************************************************************
static private int GetInt(string prompt, int min, int max)
{
    int input = 0;         //integer input by the user
    bool error = false;    //error flag, true if error

    //repeat until input is okay
    do
    {
        //Parse might throw an exception
        try
        {
            //assume no error on input
            error = false;

            //input the value
            Console.Write(prompt);
            input = int.Parse(Console.ReadLine());

            //check the value for range
            if ((input < min) || (input > max))
            {
                Console.WriteLine("An out of range value was entered.");
                Console.WriteLine("Please try again.");

                //value is out of range, set error condition true
                error = true;
            }
        }
        //catch any Exception thrown
```

Use a minimal number of parameters

```
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
            Console.WriteLine("Please try again.");

            //overflowing value was  entered, set error condition true
            error = true;
        }
    }
    //continue looping until there is no error
    while (error);

    //return the accepted integer
    return input;
}

//********************************************************************************************
//Method:    static private double GetDouble(string prompt, double min, double max)
//Purpose:   Inputs a double value with error and range checking
//Parameters:string prompt - prompt to display to the user
//           double min - minimum value accepted
//           double max - maximum value accepted
//Returns:   double - value accepted by the method
//*********************************************************************************************
static private double GetDouble(string prompt, double min, double max)
{
    double input = 0;      //double input by the user
    bool error = false;    //error flag, true if error

    //repeat until input is okay
    do
    {
        //Parse might throw an exception
        try
        {
            //assume no error on input
            error = false;
```

Method header uses this format

Parameters listed and described

Parameter names are different from variables in Main

Local variables documented using inline comments

```
            //input the value
            Console.Write(prompt);
            input = double.Parse(Console.ReadLine());

            //check the value for range
            if ((input < min) || (input > max))
            {
                Console.WriteLine("An out of range value was entered.");
                Console.WriteLine("Please try again.");

                //value is out of range, set error condition true
                error = true;
            }
        }
        //catch any Exception thrown
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
            Console.WriteLine("Please try again.");

            //overflowing value was  entered, set error condition true
            error = true;
        }
    }
    //continue looping until there is no error
    while (error);

    //return the accepted integer
    return input;
}
```

Decision and loop statements commented above the statements

Only one return point from a method

```
//********************************************************************************************
        //Method:    static private double CalculateBMI(int ft, int inch, double lbs)
        //Purpose:   Calculates body mass index
        //Parameters:int ft - height feet
        //           int inch - height inches
        //           double weight - weight in lbs.
        //Returns:   double bmi - calculated body mass index
        //*********************************************************************************************
        static private double CalculateBMI(int ft, int inch, double lbs)
        {
            double bmi = 0.0;       // body mass index
            int totalInches = 0;    // total height in inches

            // calculate the height in inches
            totalInches = ft * 12 + inch;

            // calculate the body mass index
            bmi = lbs / (totalInches * totalInches) * 703;

            // return the body mass index
            return bmi;
        }
    }
}
```

Parameter names are different from variables in Main

Local Variables documented using inline comments

All methods are private unless public needed in your library