Documentation Example 1

```
//*************************************************************************************
//Your Name Here  Assignment #
//*************************************************************************

using System;
using System.Collections.Generic;
using System.Text;

namespace ICA3
{
    class Variables
    {
        static void Main(string[] args)
        {
            int hoursWorked = 40;      //hours worked by employee
            double hourlyRate = 12.43; //hourly rate in dollars
            double total;              //total pay due
            string name;               //employee name
            string input;              //temporary storage for numeric input

            //input the employee's name
            Console.Write("Enter your name: ");
            name = Console.ReadLine();

            //input the hours worked as an integer
            Console.Write("Enter hours worked: ");
            input = Console.ReadLine();
            hoursWorked = int.Parse(input);

            //input the hourly pay rate as a double
            Console.Write("Enter your hourly rate: ");
            input = Console.ReadLine();
            hourlyRate = double.Parse(input);

            //calculate the total pay due
            total = hoursWorked * hourlyRate;

            //display the name, hours worked, hourly rate and total pay due
            Console.WriteLine("{0} here are your results...", name);
            Console.WriteLine("You worked {0} hours.", hoursWorked);
            Console.WriteLine("Your rate of pay is ${0}", hourlyRate);
            Console.WriteLine("Your pay is ${0}", total);
        }
    }
}
```

Comment block with your name and ICA

All variables defined at beginning of Main

Variables use descriptive names, and Camel case

Purpose of each variable is commented using **inline** comment

Minimal number of variables are used; note how input is used twice.

Each major block of code is separated by an empty line and commented **above** the block.

Statements do not wrap around to next line on the printout of the program

Program is properly indented as provided by the compiler.