# Lab 5 Instructions

Create a Folder called **Your Name Lab 5.**  
When you save the project below, use this folder as your location   
Include your program desigs in this folder, and then zip up the folder.**Submit the zipped folder.**  
**Project:** Complete **both project** from the following **Programming problems** at the endof **Chapter 5:**

* **Distance Calculator (problem 1)**

In addition, speed must be a decimal greater than 0 and less or equal 120

Hours must be an integer between 1 and 10 inclusive.  
Generate relevant error messages

* **Random Number File Writer (problem 13)**

In addition, the number of random numbers generated must be between 1 and 200 inclusive  
Generate relevant error messages

**Grading:**

The programming challenge is worth **62.5 points**. Points are awarded for

* A functional program that accomplishes the goal of the challenge as read in the materials (**complete the program design prior to this**.) (**30 points**)
* Proper use of loop (**10 points**)
* Proper validation (**5 points**)
* Well-designed GUI (**2.5 points**)
* Following class programming standards (**5 points**)
* Program design (**10 points**)

**Notes**:

**Follow** the standards. New this week:

* Use Combo Boxes for Input, List Boxes for Output
* Always fill Combo Boxes and List Boxes at run time, not design time

# Lab 5 – Coding Standards

**Use Combo Boxes for Input, List Boxes for Output**

**Always fill Combo Boxes and List Boxes at run time, not design time**

**Always use { } with an if statement**

if (price > DISCOUNT\_CUTOFF)

{

discountPercent = DISCOUNT\_PERCENT\_HIGH;

}

**Properly indent statements inside if statements**

if (red1RadioButton.Checked)

{

if (red2RadioButton.Checked)

{

this.BackColor= Color.Red;

}

}

**Naming standards for variables:**

Use camel case; example: tipAmount

**Named Constants**

Use named constants instead of literals in your code

Use Upper case with words separated by an underscore  
Example: TIP\_PERCENT

**Naming standards for controls:**

Control names consist of a root and a suffix

The suffix identifies the control type

The root is always mixed-case, starting with a small letter

Never use underscores in control names

|  |  |
| --- | --- |
| Control type | Example |
| Button | changeFormNameButton |
| Form | demo1Form |
| Label | helpMessageLabel |
| Textbox | customerLastNameTextBox |

You aren’t required to name identifying labels (labels not used in your code)

You can just use the default, Label1, Label2, etc. for this.

You are required to name the form, something the book doesn’t do until later.   
Get into the habit of naming forms early.

**Comment standards:**

**Every form’s code begins with a block comment section**  
that defines Project, Programmer, Date and what the form does

/\*

Project Musicians

Programmer Mike

Date March 2015

Display name of selected musician

\*/

**Every method should have a comment at the start explaining what the method does**.

These should be short so use line comments

It’s OK to use block comments if you must

private void radAniDiFranco\_Click(object sender, EventArgs e)

{

// Displays "ani diFranco" when this picture is selected

**Code should be properly indented**

private void caloriesBurnedButton\_Click(object sender, EventArgs e)

{

// verify that the calories burned per minute is integer and reasonable

// verify that minutes are selected from the combo box

// if both conditions are Ok, display the calories burned in 10 minute increments

int caloriesPerMinute, minutes, count, calories;

if (int.TryParse(caloriesPerMinuteTextBox.Text, out caloriesPerMinute) &&

caloriesPerMinute >=0 && caloriesPerMinute <=300)

{

if (minutesComboBox.SelectedIndex > -1)

{

minutes = int.Parse(minutesComboBox.SelectedItem.ToString());

caloriesBurnedListBox.Items.Clear();

for (count = 10; count <= minutes; count += 10)

{

calories = count \* caloriesPerMinute;

caloriesBurnedListBox.Items.Add(calories + " burned " + count + " minutes");

}

}

else

{

MessageBox.Show("Please select minutes from combo box");

minutesComboBox.Focus();

}

}

else

{

MessageBox.Show("Calories per minute must be an integer between 0 and 300");

caloriesPerMinuteTextBox.Focus();

caloriesPerMinuteTextBox.SelectAll();

}

}