CISC 110 Lab 2

General Instructions for All Labs/Assignments

In each of Labs 1-6, you will be completing the first portion of your assignment for that week. Once you finish your lab, you will show your work to your TA for 1% of your final mark. Then you can continue working on the rest of your assignment. Some of you may be able to finish your assignment within the lab time, but there is no rush. Your full assignment is due at the start of your next lab; you show your work to your TA for 2% of your final mark (for a total of 3% for each lab/assignment). You are also required to post your assignment in your CISC 110 folder on the Web, so that we can view everyone's assignments. Post your `.html` file and your `.swf` file in the same folder on the Web.

You will create two files: the Flash file (`.fla`) that contains your multimedia elements and the ActionScript file (`.as`) that contains your ActionScript program. When you publish your project, Flash creates the `.html` file and the `.swf` file.

Once again, do not be intimidated by the code that you don't understand. By the end of the course, you will understand everything in this template, but for now you will use the template and add code to specific parts. Also, remember to save your work often.

After you finish your lab, you may leave if you wish, but it's recommended to stay and work on the rest of your assignment in order to be able to ask the TA questions. Also ask questions of each other and share your knowledge!

Specific Instructions for Lab 2

In this lab, you will complete the first portion of Assignment 2. In your assignment, you will create an Hourly Wage Calculator, similar to the wages application on page 75 of your text. It will tell you what your hourly wage was for any year, once you type in your yearly income, the number of weeks you worked that year, the number of days you worked per week, and the number of hours you worked per day. We're assuming a job in which you work the same number of hours each work day and the same number of days each work week.

In the lab portion, you will create a simplified version. The simplified Hourly Wage Calculator will calculate an hourly wage assuming 50 standard work weeks : 8 hours per day, 5 days per week. Since 50 * 8 *5 = 2000, that's 2000 work hours in one year. Therefore, the formula for the hourly wage will be: hourlyWage = yearlyIncome / 2000. For instance, if the yearly income is $40,000, the hourly wage is $20.00. Equivalently, the formula could be: hourlyWage = yearlyIncome / (50 * 8 * 5). Notice that the numbers being multiplied must be inside parentheses. Otherwise the yearly income would only be divided by 50 instead of by 2000.

Here are the required data objects:

Object Identifier	Description	Data Type
`yearlyIncome`	Variable data object that stores the yearly income input by the user	Number
`hourlyWage`	Variable data object that stores the computed hourly wage based on 50 weeks of 5 8-hour days Computation: hourlyWage = yearlyIncome / (50 * 8 * 5)	Number

Here are the required visual objects:

Object Identifier	Description	Type of Object
`InputIncome`	Where the user inputs text that will be converted to a Number and stored in the variable `yearlyIncome`	Input text field
`ComputeBtn`	Where the user activates the computation process	Button
`Output`	Used to display the hourly wage	Dynamic text field

Here are the steps for you to complete:

1. Create a new folder called `Assign2` in your CISC 110 folder. Then create a new ActionScript 3.0 file called `Assign2.fla` and save it in your `Assign2` folder (`Create New | ActionScript 3.0`).
2. Add a Static Text Field somewhere on the stage with the text “Yearly Income”. Decide on a Font you will use for all of your text, for instance Arial Bold. With the Properties Inspector open, select `Embed` and then under `Character Ranges`, select `Uppercase`, `Lowercase`, `Numerals, and Punctuation`. It’s best to embed all fonts that you use in text fields, so they will display properly on any computer. Embedding a font adds its definition to your library, so then you can use it in as many text fields as you wish.
3. Add an Input Text Field next to it, in which the user will type their yearly income, with the initial text: “0”. Specify the instance name of the text field as `InputIncome`. The ActionScript program will reference this text field by this instance name.
4. Add a Dynamic Text Field at the bottom to display the hourly wage, once it’s calculated. Specify the instance name of the text field as `Output`. The ActionScript program will reference this text field by this instance name.
5. Add a button. You could follow the same instructions you used in Lab 1, on p. 41 of your text, but change the text of the button to “Calculate” and specify its instance name as “ComputeBtn”.

6. Specify your document class to be `Assign2,` similar to p. 43 of your text. This tells your Flash file (`.fla`) to link to your ActionScript file (`.as`).
7. Download the file `Assign2.as` from the CISC 110 website under Lab 2. This contains a starting template for your program script.
8. Make the changes below to your `Assign2.as` file. After each change, save your .as file and run your movie. If you don't save it first, you will be running the old version. Use `trace` statements wherever useful to check the values of variables. Look at `wagesApp.as` on page 79 and 80 of your text to see a very similar example.
9. Under the comment, `// TASK 1: Variable Declarations`, define two new Number variables, one called `yearlyIncome` and one called `hourlyWage`.
10. Under the comment, `//TASK 2: Gather Input`, assign `yearlyIncome` to be the numeric value of the String in the `InputIncome` text field. Since text fields contain String values, you must convert the value in `InputIncome.text` to a Number using the Number method. The user must type in an income with no dollar sign and no comma.
11. Under the comment, `// TASK 3: Compute the Hourly Wage`, calculate the hourly wage using the formula given in the description at the start of this lab.
12. Under the comment, `// TASK 4: Display the Hourly Wage`, display the hourly wage computed in the last step in the dynamic text field `Output`. Use the `toFixed` method to convert the numeric value to a String and display it with two decimal places.
13. Run your movie, input a yearly income with no comma and no dollar sign, and try out the button.

Lab 2 Marking Scheme (1% of final mark)

Marked out of 5:

1 mark - One Input text field, named `InputIncome`, set to hold an initial value of 0

1 mark - One dynamic text field, named `Output`, to display the hourly wage

3 marks - One button, named `ComputeBtn`, that causes the following:

A. the yearly income to be read from the `InputIncome` text field
B. the hourly wage to be calculated from the yearly income
C. the hourly wage to be displayed in the `Output` text field