

# ITEC 136 Lab 1 – 2010 Winter

## Purpose

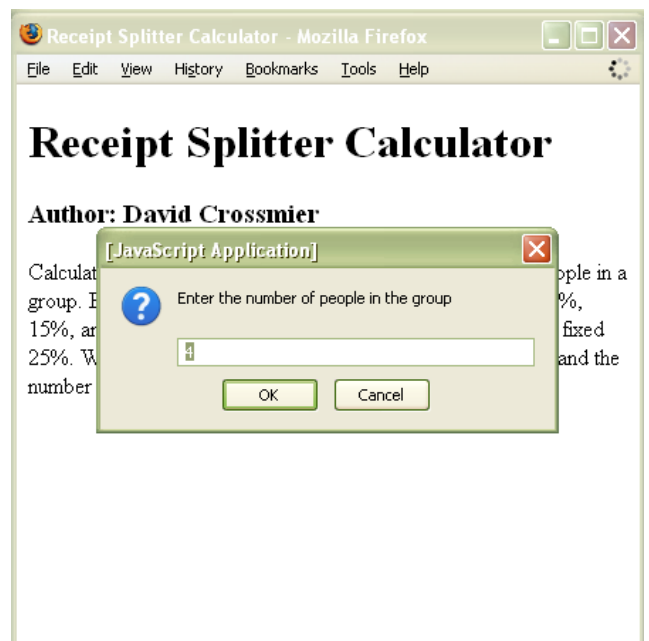
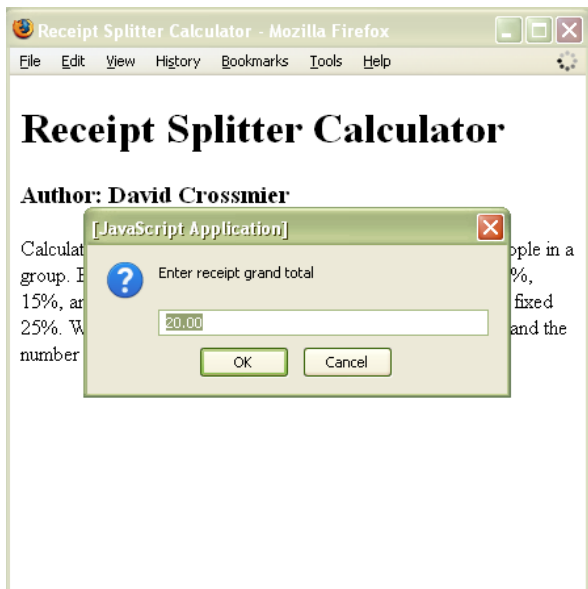
To assess your ability to apply the knowledge and skills developed in Module 1, 2 and Module 3. Emphasis will be placed on the following learning outcomes:

1. Create and display simple syntactically correct HTML documents.
2. Describe the execution a JavaScript program in a web browser.
3. Describe arithmetic, relational, and logical operators in terms of their input and output data types.
4. Declare, define, and use variables in a script.
5. Correctly use the JavaScript conditional "if" statement.

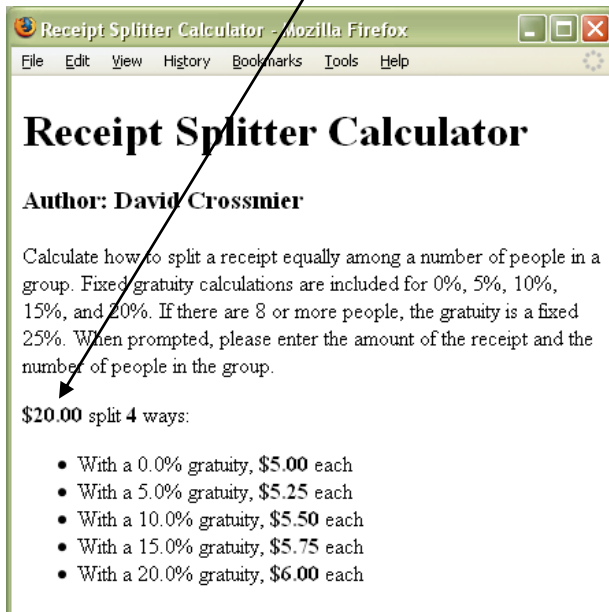
## Assignment

A number of friends decide to go out to eat at a family-style restaurant where all menu items are a-la-carte. The group decides to order food for the table and split the bill equally among the people in the group. Your job in this assignment is to write a receipt-splitter and calculator that is given the input of a receipt amount and a number of guests and whose output is the price per person. Sometimes the service is excellent, and sometimes it is poor. Thus, your program should output a price per person based on several fixed gratuity amounts (0%, 5%, 10%, 15% and 20%). Additionally, if there are 8 or more guests, do not output the fixed gratuity amounts, but output a fixed gratuity of 25% only. Make sure you put your name and the assignment number either on the web page or in comments. **You must use at least one function as described below in your solution.**

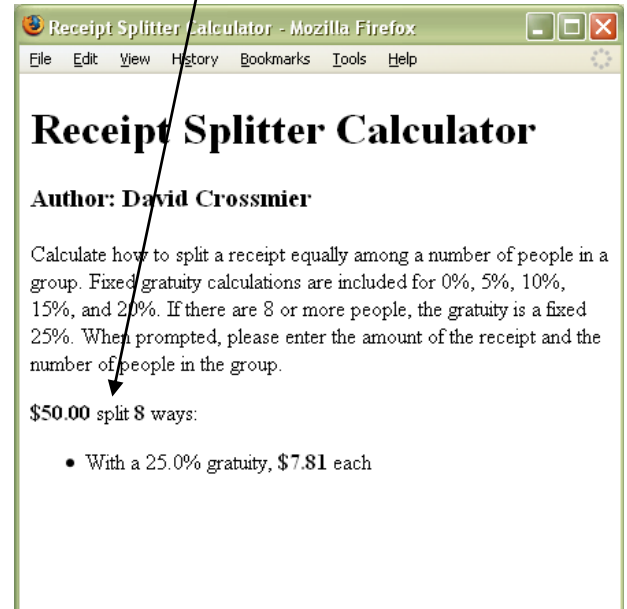
Some example screen shots (in Firefox) of the application in action are given below:



Note that total and guest count is dynamically written based on prompt() values.



I used F5 to reload the page and gave \$50 for grand total and 8 people to test a large group. Note that only 25% should print.



## Additional Action Items/Notes

- You must use a function in your JS file to calculate to output the tip percentages and per person costs bullet items. The function should have two parameters, the number of guests, and the amount of the bill.
- All significant JavaScript should be in an external file with a ".js" extension. Remember to put the HTM and JS file in the same folder (directory) so that the JavaScript is found without additional efforts. Typically, the prompts and a call to your function will be in the HTML within <script> tags.
- The prompt() function must be used to all gather information from the user just as I have done in the example screen shots.
- Try to make your solution look as close to the above screen shots as possible.
- Review code against the documentation and style requirements Word document available on the course web site for the assignment. You will find that both documentation and style requirements are facilitated by the Aptana Integrated Development Environment (IDE). For example, when you create a new untitled JavaScript file, by default a comment block is generated, possibly with your log in name. Please put your full name, course name and assignment number in the comments of each text based file you deliver.
- You should name your HTML file either index.htm or index.html, either is fine.
- No need to validate your code with the W3C validator, <http://validator.w3.org/> as the first homework required.

## Helpful Hints

The following hints may help you to solve the problem:

1. **On my computer, the preview tab/pane for Firefox in Aptana does not appropriately display the prompt() JavaScript calls. It seems only the HTML renders without the prompts. If I used the “Run Firefox – Internal Server” toolbar button or loaded the page manually in Firefox, it works fine. Since the assignment is graded using Firefox, I suggest you use the green toolbar button that resembles a VCR play button when testing the assignment.**

2. Use the prompt() function to input data. Recall that this returns a string value. For example:

```
var favoriteColor = prompt("What is your favorite color?", "yellow");
```

The first parameter is shown to the user and the second parameter is the default response.

3. Use the parseFloat() function to convert a string into a floating point number. After the JavaScript below, receiptTotal will hold a floating point value with decimal precision, ready to be used in math operations.

```
var numberAsString = "6.023E23";  
var receiptTotal = parseFloat(numberAsString);
```

4. Use the parseInt() function to convert a string into an integer number. After the JavaScript below, totalPeople will hold a whole number value of 123, ready to be used in math operations.

```
var totalPeopleAsString = "123";  
var totalPeople = parseInt(totalPeopleAsString);
```

5. Use the toFixed() method of any Number object to convert a number into a string with a fixed number of decimal places. For example:

```
var myPI = 3.1415926535;  
document.writeln(myPI.toFixed(4)); // writes out 3.1416
```

6. Use Math.round to round a number to the nearest integer

```
var wholeNum = Math.round(20.49); // Returns 20
```

## Grading Criteria (also on lab assignment web page)

Lab assignments will be graded in at least the following areas:

- Correct results (0 to 40 points) – the program should work, and be seen to work. This involves not only producing correct output, but also reacting well to user error, incorrect input, etc. The code should be *robust*.
- Style standards (0 to 5 points) – It is not uncommon for industry to impose a *way* of writing software that is uniform across all programmers. As listed in the action items of the course web page lab assignment, ensure that your solution code meets the documentation and style guidelines

for this course. The documentation and style requirements Word document (DocumentationAndStyleGuidelines.doc”) exists on the lab assignment page on the course web site and in the Resources folder on your course CD. Review your code against this document. I am looking for consistency also. For example, if you prefer to put your { on the same line as the *if* statement rather than the next line, that is fine provided you are consistent throughout your code.

- Documentation standards (0 to 5 points) – It is not uncommon for industry to impose a *way* of documenting software that is uniform across all programmers. Documentation standards are outlined in the same document referenced in the style standards above.