Laboratory 4

Objective
In this week's laboratory we will investigate the use of variables in computer software. We will create and manipulate the three main types (numbers, text and boolean) of variables. We will look at the main functions provided by Scribble to provide input and output.

The aim of the laboratory is to help you:
1. Learn how to use IF-THEN decision logic in your Scribble applications,
2. Use IF-THEN decision logic in combination with the built-in IS-A operator to check end-user data input and ensure correct operation of applications you make, and
3. Consolidate your understanding of the link between flowchart and decision table representations of decision logic and Scribble code.

Required resources
To perform this laboratory at Monash:
• You will need a Windows-based PC in a Monash student laboratory
• Your authcate credentials to login to the computer and to the unit-web site on Moodle

To perform this laboratory at home on your own computer:
• A working copy of Scribble
• Your authcate credentials to login to the computer and to the unit-web site on Moodle
Tasks (on- and off-campus students)

1. Numbers in Scribble: A pay calculating program revisited.

   In last week’s laboratory session you made an application that implemented the flowchart on the left. The code you used follows. This week we are going to modify that application (if you didn’t keep a copy you can download one from the unit web site).

   The modification we are going to make will alter the program to “trap” poor quality input. We will check that the user has in fact typed in a positive number before making the calculation. The following code implements the modification. Note the use of IF-THEN-ELSE statements in conjunction with the logical operator AND and the IS-A NUMBER operator.
Make that program, and check that it runs as you expected. Test it with a variety of test data. Include some nonsense values.

2. A calculating a discount sale price. Last week you made a Scribble application that implemented the following flowchart. Modify that application using the technique above (again a copy of the application can be downloaded from the unit web site if you have’t kept a copy).

![Flowchart for calculating a discount sale price]

3. Implementing the tutorial tasks.

3.1. BMI Index. Create a Scribble application that asks the user for their mass in kgs and their height in m. That information should be used to then calculate that individual's BMI (using the formula \( \text{BMI} = \frac{\text{mass}}{\text{height}^2} \)). A classification of the individual's BMI should then be displayed along with the BMI value. The following table shows the categories that are used to classify the BMI (body mass index) of an adult. Use the information in this table to create a flowchart for a program that asks a user for the BMI, and then displays the category that they fall into. (This table is taken from the wikipedia page on BMI: [http://en.wikipedia.org/wiki/Body_mass_index](http://en.wikipedia.org/wiki/Body_mass_index).

<table>
<thead>
<tr>
<th>Category</th>
<th>BMI range – kg/m²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very severely underweight</td>
<td>less than 15</td>
</tr>
<tr>
<td>Severely underweight</td>
<td>from 15.0 to 16.0</td>
</tr>
<tr>
<td>Underweight</td>
<td>from 16.0 to 18.5</td>
</tr>
<tr>
<td>Normal (healthy weight)</td>
<td>from 18.5 to 25</td>
</tr>
<tr>
<td>Overweight</td>
<td>from 25 to 30</td>
</tr>
<tr>
<td>Obese Class I (Moderately obese)</td>
<td>from 30 to 35</td>
</tr>
<tr>
<td>Obese Class II (Severely obese)</td>
<td>from 35 to 40</td>
</tr>
</tbody>
</table>
3.2. **Extending the BMI Index calculator.** Use IF-THEN decision logic to check the input’s to the BMI index are valid and reasonable. Don’t allow any incorrect or weird data to be input. If that happens provide the end-user with an appropriate error message.

3.3. **Growth share classification application.** Create a Scribble application that implements the BCG growth share classification. The application should ask a user to input the annual market growth of a product and the product’s relative market share. Once those inputs have been obtained your application should classify the product as a Star, Question mark, Cash cow or Dog and display that classification to the user.

Market growth is measured as a percentage of the size of the current year’s market compared to last year’s market (the market is measured in overall revenue). Relative market share is measured as a ratio of the market share of the product compared to the largest competitor. A value of 1 means the products share is equal to the largest competition. Less that one means that if trails the largest competitor; while a value greater than 1 means the product is the market share leader in the market. If a product is in a market that has annual growth that is greater than 10% per annum and it has a relative market share of greater than one it is classified as a “Star”. If it is in a market that has annual growth that is greater than 10% per annum but has a relative market share less than 1 then the product is classified as a “Question Mark”. If a product is in a market that has less than 10% annual growth but has relative market share greater than 1, then it is classified as a “Cash cow”. If it is in a market that has less than 10% annual growth and has a relative market share less than 1, then it is classified as a “Dog”.


4. If you have time left ... [download the Pong game](#) demonstrated in lectures and work on improving it.

5. **On-campus students.** That’s it - with about 30 minutes to go in the class your tutor will distribute the in-class test for this laboratory. This is worth 1% of your mark for the unit. It should only take 15 to 20 minutes to complete. You must work on it on your own. Give your answer sheet back to your tutor at the end of the class. You will get the result next week (don't worry, it's not hard).

5. **Off-campus students.** On the “study guide” tab of the unit-web site for this week you will find a link to this week’s on-line test for the semester. This is worth 1% of your mark for the unit. It should only take 15 to 20 minutes to complete. You must work on it on your own. The test will be available for you to do for 1 week (you won’t be able to access it after midnight on Sunday). If you have any problem accessing the test please send Peter O’Donnell (peter.odonnell@monash.edu) an email.

---

**Further reading and information**

Wikipedia entries on: