INF 154
PRACTICAL 3
2023

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

At the end this practical you should be able…

1. To analyse a problem and create an algorithm.
2. To design your interface.
3. To structure your code.
4. To convert your algorithm into code.

What are we going to do?

- Prac 3a: Completed in class.
- Prac 3b: Completed in class.
- Prac 3c: Take home practical.

Practical Exercise 3a

Objective: We would like to create a program where the user can calculate how much they would weigh on the Moon, Mars or Jupiter.

Practical Exercise 3a

Key questions:

- Inputs and Obtaining Data:
  - How are we obtaining the user's information?
    - » With regards to the weight: is it via a textbox or numeric up down-> What does this mean for the data type?
    - » How is the user selecting their different options?
  - Once I have the interface to obtain the data, do I need to manipulate it?
- Processing:
  - How do I solve the problem?
    - » What formulas do I need?
    - » What steps are required?

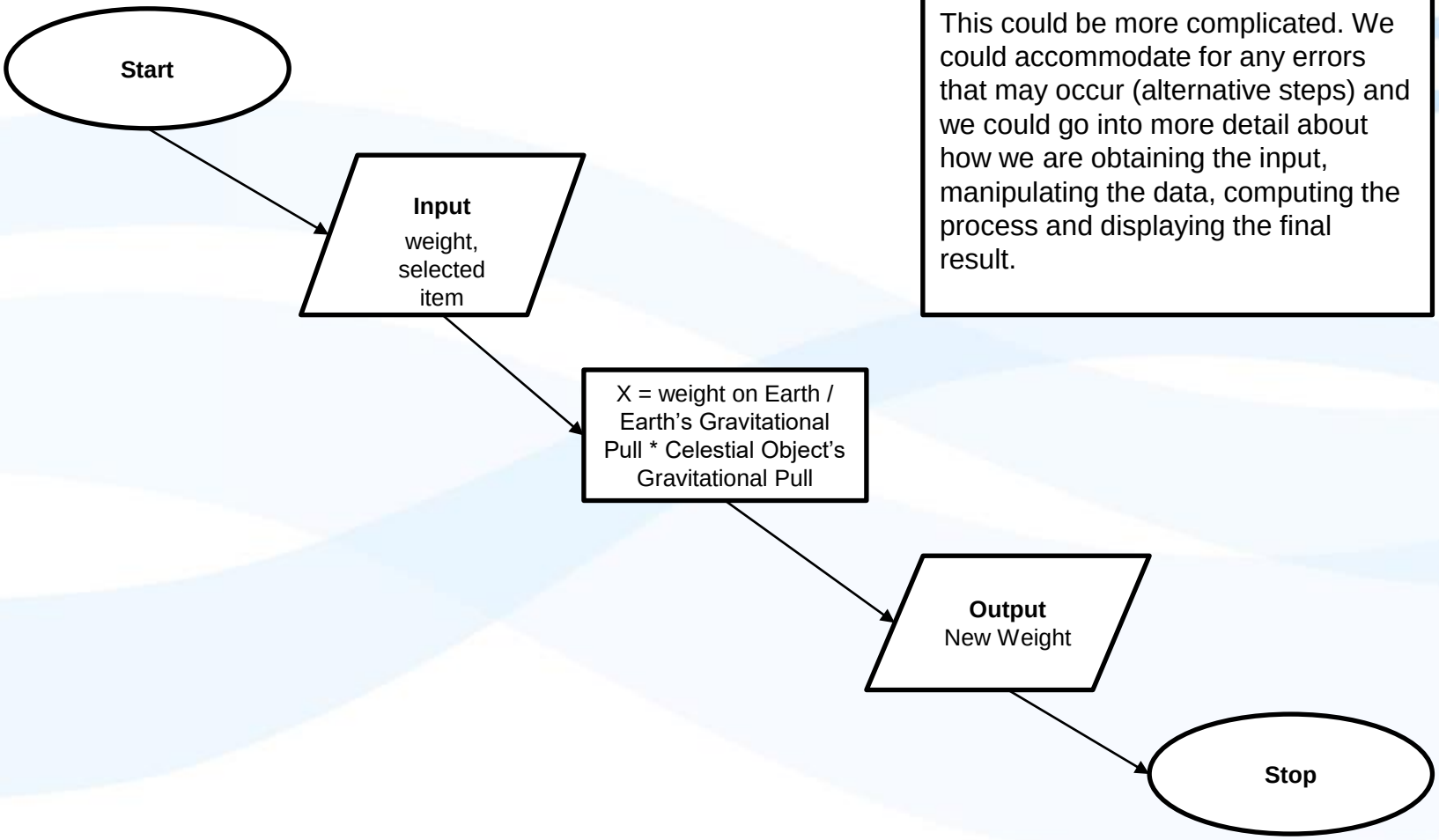
Practical Exercise 3a

Key questions:

- Outputs:
  - What type of information is going to be displayed back to the user?
    - » Does it have to be manipulated again?
    - » Must all of it be displayed to the user?
    - » Is it via a textbox or numeric up down-> What does this mean for the data type?

Practical Exercise 3a

The algorithm:



Note:

This could be more complicated. We could accommodate for any errors that may occur (alternative steps) and we could go into more detail about how we are obtaining the input, manipulating the data, computing the process and displaying the final result.

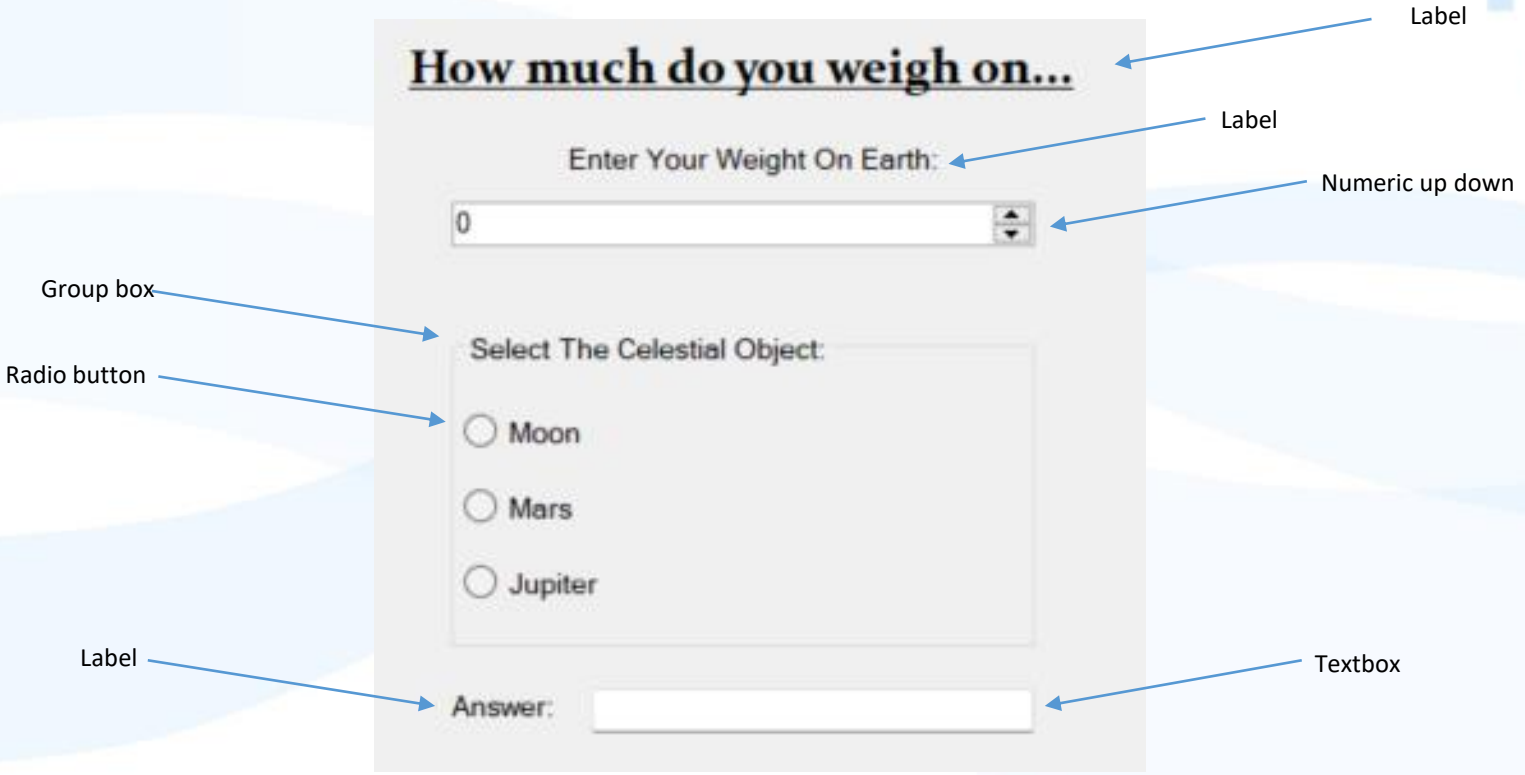
Practical Exercise 3a

Input *(Capturing)*	Processes	Output
 Numeric up down: • Dealing with a number. Therefore, double or int.  Radio button: • When selecting an option, a value needs to be associated with it. Is this value going to change anytime soon? Gravitational pull, no. It's a constant.	X = weight on Earth / Earth's Gravitational Pull * Celestial Object's Gravitational Pull • double inputtedValue. • const double. EARTH_GRAVITATIONAL_PULL. • const double CELESTIAL_OBJECT_GRAVITATIONAL_PULL. X = inputtedValue / EARTH_GRAVITATIONAL_PULL * CELESTIAL_OBJECT_GRAVITATIONAL_PULL.	 Textbox. Note: Also consider the data types that will be required when inputting, processing and outputting.

Practical Exercise 3a

The interface:

Now that we have an understanding of the program we would like to implement. We can design the interface.



Practical Exercise 3a

Guideline for setting up your code:

- Declare your variables.
- Get your inputs and deal with data manipulation.
- Processing & Calculations
- Deal with data manipulation and display the results (output).

Example from lecture 3:

```
private void btnCalculate_Click(object sender, EventArgs e)
{
    // Declare variables

    // Get input and convert to integer

    // Calculate total kms

    // Calculate total amount

    // Convert total amount to string and display in message box
}
```

Practical Exercise 3a

The code:

– Declaring variables:

```
20  // Source for constants: https://www.bbc.co.uk/bitesize/guides/zg638mn/revision/2
21  // Formula: x = Weight on Earth / Earth's Gravitational Pull * Celestial Object's Gravitational Pull
22
23  // Declare variables
24  double inputtedMass;
25  string finalAnswer;
26  const double MOON_GRAVITATIONAL_PULL = 1.7;
27  const double EARTH_GRAVITATIONAL_PULL = 9.8;
28  const double MARS_GRAVITATIONAL_PULL = 3.7;
29  const double JUPTER_GRAVITATIONAL_PULL = 24.7;
```

Note:

– Notice how the variables and constants are not declared within an event handler. Because we will be using them in all the different event handlers, we declared global variables and constants. Now we only need to declare them once.

Practical Exercise 3a

The code:

– Get your inputs and deal with data manipulation.

```
31  private void radMoon_CheckedChanged(object sender, EventArgs e)
32  {
33      // Get inputs
34      inputtedMass = Convert.ToDouble(nudWeight.Value);
35      // Processes/Calculations
36      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * MOON_GRAVITATIONAL_PULL);
37      // Output
38      txtAnswer.Text = finalAnswer;
39  }
```

```
41  private void radMars_CheckedChanged(object sender, EventArgs e)
42  {
43      // Get inputs
44      inputtedMass = Convert.ToDouble(nudWeight.Value);
45      // Processes/Calculations
46      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * MARS_GRAVITATIONAL_PULL);
47      // Output
48      txtAnswer.Text = finalAnswer;
49  }
```

```
51  private void radJupiter_CheckedChanged(object sender, EventArgs e)
52  {
53      // Get inputs
54      inputtedMass = Convert.ToDouble(nudWeight.Value);
55      // Processes/Calculations
56      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * JUPTER_GRAVITATIONAL_PULL)
57      // Output
58      txtAnswer.Text = finalAnswer;
59  }
```

Practical Exercise 3a

The code:

– Processing & Calculations.

```
31  private void radMoon_CheckedChanged(object sender, EventArgs e)
32  {
33      // Get inputs
34      inputtedMass = Convert.ToDouble(nudWeight.Value);
35      // Processes/Calculations
36      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * MOON_GRAVITATIONAL_PULL);
37      // Output
38      txtAnswer.Text = finalAnswer;
39  }
```

```
41  private void radMars_CheckedChanged(object sender, EventArgs e)
42  {
43      // Get inputs
44      inputtedMass = Convert.ToDouble(nudWeight.Value);
45      // Processes/Calculations
46      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * MARS_GRAVITATIONAL_PULL);
47      // Output
48      txtAnswer.Text = finalAnswer;
49  }
```

```
51  private void radJupiter_CheckedChanged(object sender, EventArgs e)
52  {
53      // Get inputs
54      inputtedMass = Convert.ToDouble(nudWeight.Value);
55      // Processes/Calculations
56      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * JUPTER_GRAVITATIONAL_PULL);
57      // Output
58      txtAnswer.Text = finalAnswer;
59  }
```

Practical Exercise 3a

The code:

– Deal with data manipulation and display the results (output).

```
31  private void radMoon_CheckedChanged(object sender, EventArgs e)
32  {
33      // Get inputs
34      inputtedMass = Convert.ToDouble(nudWeight.Value);
35      // Processes/Calculations
36      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * MOON_GRAVITATIONAL_PULL);
37      // Output
38      txtAnswer.Text = finalAnswer;
39  }
```

```
41  private void radMars_CheckedChanged(object sender, EventArgs e)
42  {
43      // Get inputs
44      inputtedMass = Convert.ToDouble(nudWeight.Value);
45      // Processes/Calculations
46      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * MARS_GRAVITATIONAL_PULL);
47      // Output
48      txtAnswer.Text = finalAnswer;
49  }
```

```
51  private void radJupiter_CheckedChanged(object sender, EventArgs e)
52  {
53      // Get inputs
54      inputtedMass = Convert.ToDouble(nudWeight.Value);
55      // Processes/Calculations
56      finalAnswer = Convert.ToString(inputtedMass / EARTH_GRAVITATIONAL_PULL * JUPTER_GRAVITATIONAL_PULL)
57      // Output
58      txtAnswer.Text = finalAnswer;
59  }
```

Practical Exercise 3b

Objective: We would like to create a program where the user enters two numbers and selects one of the following options; addition, subtraction, multiplication, modulus and division. Once two numbers have been provided and an option has been selected, the result will be displayed. We would also like to allow the user to clear the results from the interface.

Practical Exercise 3b

Key questions:

- Inputs and Obtaining Data:
  - How are we obtaining the user's information?
    - » With regards to the numbers: is it via a textbox or numeric up down-> What does this mean for the data type?
    - » How is the user selecting their different options?
  - Once I have the interface to obtain the data, do I need to manipulate it?
- Processing:
  - How do I solve the problem?
    - » What formulas do I need?
    - » What steps are required?

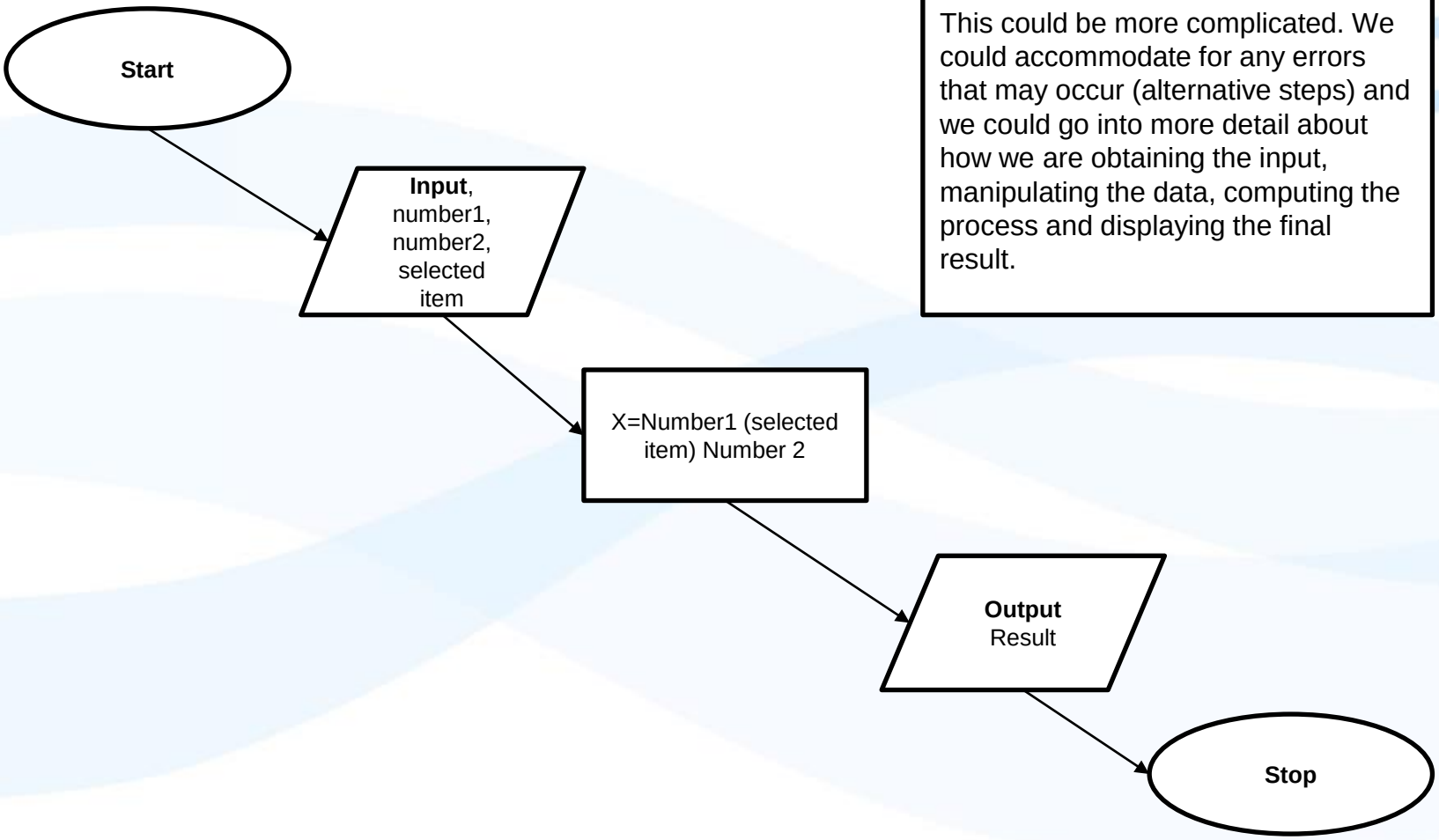
Practical Exercise 3b

Key questions:

- Outputs:
  - What type of information is going to be displayed back to the user?
    - » Does it have to be manipulated again?
    - » Must all of it be displayed to the user?
    - » Is it via a textbox or numeric up down-> What does this mean for the data type?

Practical Exercise 3b

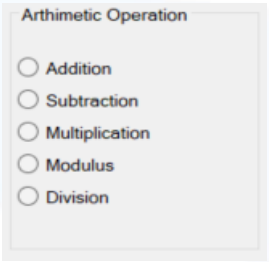
The algorithm:



Note:

This could be more complicated. We could accommodate for any errors that may occur (alternative steps) and we could go into more detail about how we are obtaining the input, manipulating the data, computing the process and displaying the final result.

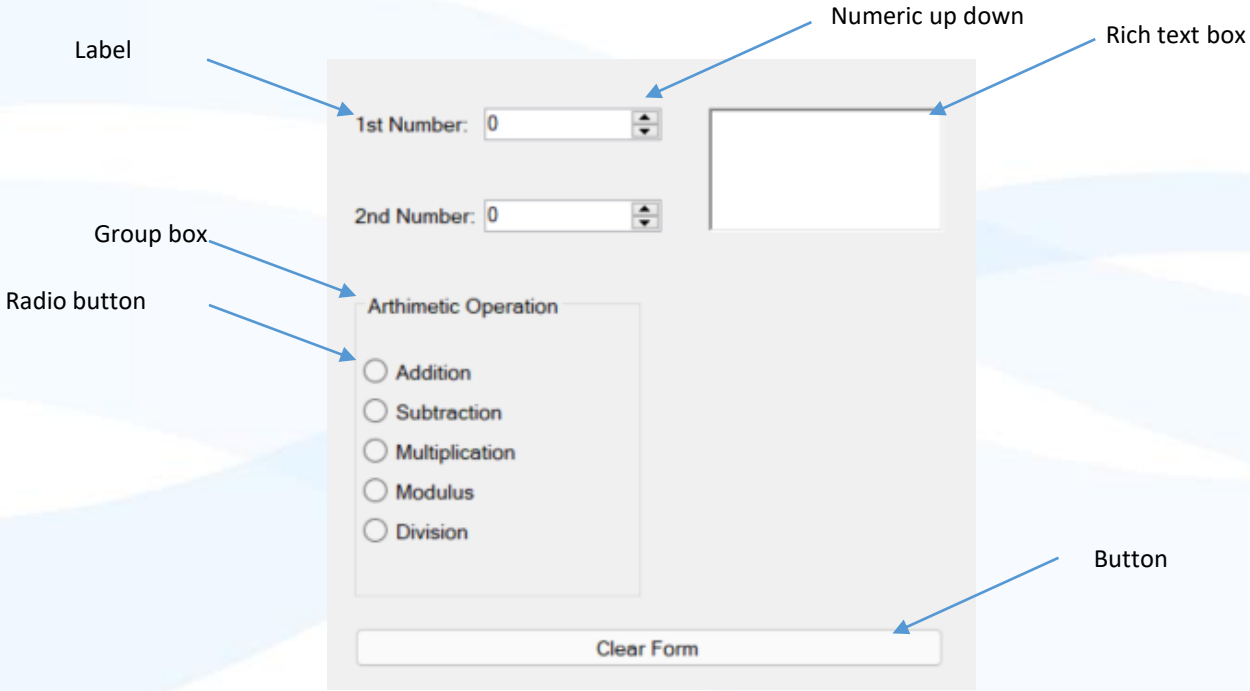
Practical Exercise 3b

Input *(Capturing)*	Processes	Output
 Numeric up down: • Dealing with a number. Therefore, double or int.  Radio button: • When selecting an option, a specific and different process will occur for each item. Therefore the process/calculations will be different.	Addition: X = number1 + number2 Subtraction: X = number1 - number2 Multiplication: X = number1 * number2 Modulus: X = number1 % number 2 Division: X = number1 / number 2	 Rich text box. Note: Also consider the data types that will be required when inputting, processing and outputting.

Practical Exercise 3b

The interface:

Now that we have an understanding of the program we would like to implement. We can design the interface.



Practical Exercise 3b

The code:

- Declaring variables:

```
18          }
19          // Delcare variables
20          double noOne, noTwo, finalAnswer;
21
```

Note:

- Notice how the variables and constants are not declared within an event handler. Because we will be using them in all the different event handlers, we declared global variables and constants. Now we only need to declare them once.

Practical Exercise 3b

The code:

- Get your inputs and deal with data manipulation.

```
22      private void radSubtraction_CheckedChanged(object sender, EventArgs e)
23      {
24          // Get inputs
25          noOne = Convert.ToDouble(nudNoOne.Value);
26          noTwo = Convert.ToDouble(nudNoTwo.Value);
27          // Processes/Calculations
28          finalAnswer = noOne - noTwo;
29          // Output
30          rtbForm.Text = Convert.ToString(finalAnswer);
31      }
```

```
22      private void radSubtraction_CheckedChanged(object sender, EventArgs e)
23      {
24          // Get inputs
25          noOne = Convert.ToDouble(nudNoOne.Value);
26          noTwo = Convert.ToDouble(nudNoTwo.Value);
27          // Processes/Calculations
28          finalAnswer = noOne - noTwo;
29          // Output
30          rtbForm.Text = Convert.ToString(finalAnswer);
31      }
```

Practical Exercise 3b

The code:

- Get your inputs and deal with data manipulation.

```
22      private void radSubtraction_CheckedChanged(object sender, EventArgs e)
23      {
24          // Get inputs
25          noOne = Convert.ToDouble(nudNoOne.Value);
26          noTwo = Convert.ToDouble(nudNoTwo.Value);
27          // Processes/Calculations
28          finalAnswer = noOne - noTwo;
29          // Output
30          rtbForm.Text = Convert.ToString(finalAnswer);
31      }
```

```
66      private void btnClear_Click(object sender, EventArgs e)
67      {
68          //rtbForm.Clear();
69          rtbForm.Text = "";
70      }
```

Practical Exercise 3c (Part 1 & 2)

Practical assignment to try at home:

- Part 1:
  - You will need to create a flow chart to help you decide whether you should buy a new textbook or rather buy a used textbook, depending on the textbook's depreciated value after a year.
- Part 2:
  - You will need to code a program to help you save money by creating a dashboard of your Tax-Free Savings. South Africa allows for a R500 000 lifetime limit in Tax-Free savings, with a limit of R36 000 contribution per year. Your dashboard will show all your contributions and key figures.

NOTE: Zip your code submission and Flow chart together **OR** upload your Zip and Flow chart separately as **ONE** submission. (View the screenshot on the following slide of how to submit two files)

Follow the instructions carefully…

Practical 3 Submission

Submit your Practical 3c project on ClickUP as follows:

- **Due Date: 20th March 2023.**
- Name your project, **INF154Prac3xxxxxxxxx** (where xxxxxxxxx is your student number) and compress (zip) your project.
- Your Flow chart submission needs to be a pdf, jpg or png file.
- Submit it under the Practical 3 submission link.

If you are submitting two separate files, your submission should look as follows.