

INF 154 PRACTICAL 4 2023



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihalefi

WARNING!!!

- After Practical 4, we will not be marking work done on student laptops, **only work downloaded via ClickUP** will be marked (We strongly advise you to re-watch the upload-videos on how to successfully upload homework assignments if you are still struggling with uploading your assignments)
- If you cannot attend the practical classes for your assignments to be marked, you should email us well in advance with a valid excuse, and after we evaluate, we can come up with a solution.
- In other words, **ALWAYS TRY TO ATTEND THE PRACTICAL CLASSES** if you want your assignments to be marked



At the end this practical you should be able...

1. To use if statements for conditional execution in your programs
2. To use nested if-statements and else-if statements
3. To write conditional statements using conditional operators such as ==,<,>,>=
4. To use logical operators &&, || and !
5. To do basic error handling

What are we going to do?

- Prac 4a: Mass Converter(Completed in class)
- Prac 4b: Grade Verifier (Completed in class)
- Prac 4c: Take home practical.



Practical Exercise 4a

Objective: In this practical section, you are tasked with creating a windows form program which will convert ***kilograms*** to ***pounds*** or ***grams*** depending on what the user selected.

The conversion rates are as follows

- ❖ 1 kilogram is equal to 2.20462262 pounds
- ❖ 1 kilogram is equal to 1000 grams

Practical Exercise 4a

Expected Screen Design

Create the following form with the appropriate controls. Remember to name your controls accordingly (e.g for a button you would use btnConvert).

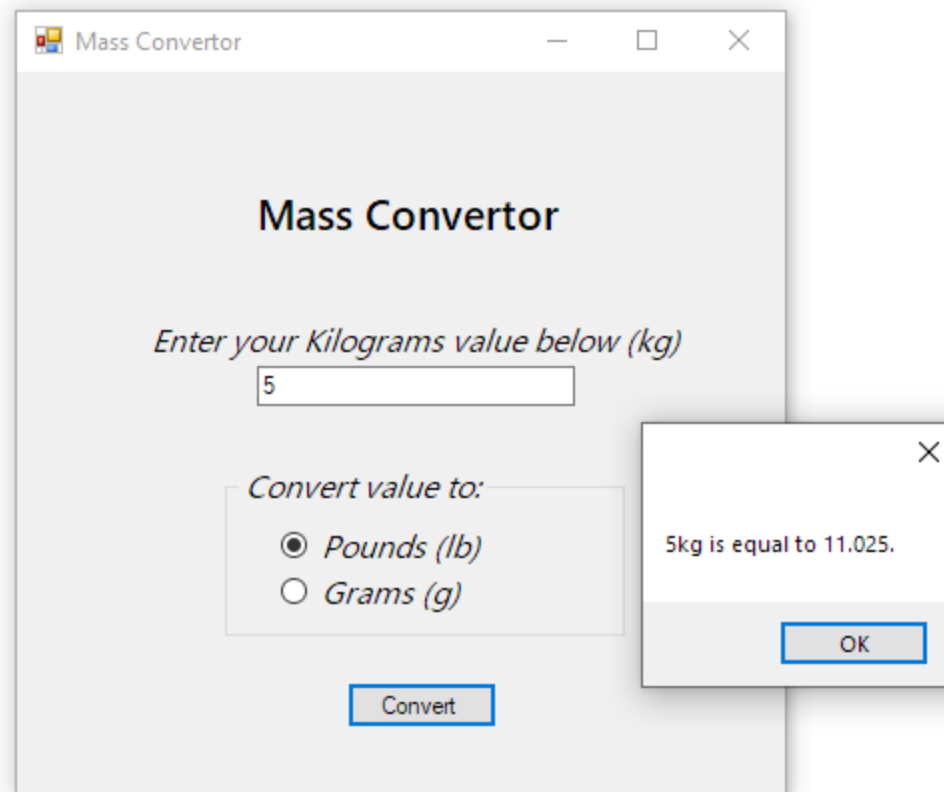
The diagram illustrates a 'Mass Converter' form with the following components and their corresponding labels:

- Label:** Points to the title 'Mass Converter' at the top of the form.
- Textbox:** Points to the input field for entering a value, with the prompt 'Enter your Kilograms value below (kg)' above it.
- Radio buttons:** Points to the selection options under the heading 'Convert value to:', which include 'Pounds (lb)' and 'Grams (g)'.
- Button:** Points to the 'Convert' button at the bottom of the form.

Practical Exercise 4a

Expected Functionality

When the convert button is clicked and the pounds radio button is selected, A message box will display the result as follows.



Practical Exercise 4a

Solution

```
private void btnConvert_Click(object sender, EventArgs e)
{
    // variables
    double inputKG;
    const double POUNDS_CONVERSION_RATE = 2.205;
    const double GRAMS_CONVERSION_RATE = 1000;
    double output;

    // input
    inputKG = Convert.ToDouble(txtKilograms.Text);

    // process
    if (radPounds.Checked == true)
    { output = inputKG * POUNDS_CONVERSION_RATE; }
    else
    { output = inputKG * GRAMS_CONVERSION_RATE; }

    // output
    MessageBox.Show(inputKG + "kg is equal to " + output + ".");
}
```



Practical Exercise 4b

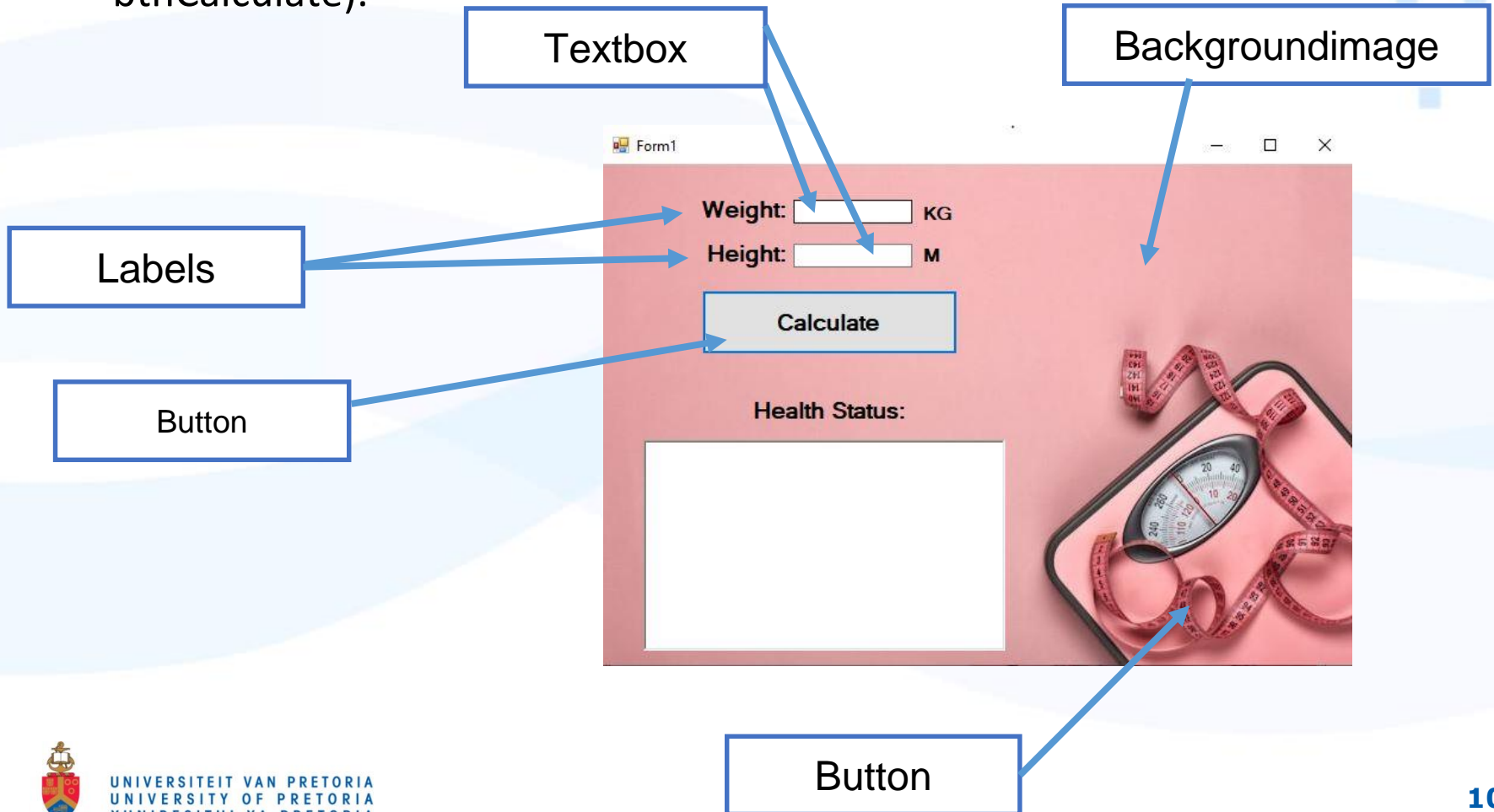
Objective: In this practical section, you are tasked with creating a windows form program which will capture the BMI of the user. Upon capturing the two required inputs, the program should calculate the BMI amount and indicate whether the user is either underweight, normal, overweight or obese.



Practical Exercise 4b

Expected Screen Design

Create the following form with the appropriate controls. Remember to name your controls accordingly (e.g for the button you would use btnCalculate).



Practical Exercise 4b

BMI FORMULA:

$$\text{BMI} = \text{Weight} / (\text{Height})^2$$

BMI	Status
≤ 18.4	Underweight
18.5 - 24.9	Normal
25.0 - 39.9	Overweight
≥ 40.0	Obese



Practical Exercise 4b

Expected Functionality

When the Calculate button is clicked, the BMI percentage should be displayed in the richtextbox and the color of the textbox should change based on the BMI percentage (indicating the status):

Form1

Weight: 55 KG
Height: 1,77 M

Calculate

Health Status:

Your BMI is: 17,5556193941715
Status: Underweight

This screenshot shows the BMI calculator interface with a pink background. The weight is 55 KG and height is 1,77 M. The 'Calculate' button is a light blue rectangle. Below it, the 'Health Status' section is highlighted with an orange background, displaying the BMI value and the status 'Underweight'. A red measuring tape and a round analog scale are visible on the right side of the form.

Form1

Weight: 65 KG
Height: 1,77 M

Calculate

Health Status:

Your BMI is: 20,7475501931118
Status: Normal

This screenshot shows the BMI calculator interface with a pink background. The weight is 65 KG and height is 1,77 M. The 'Calculate' button is a light blue rectangle. Below it, the 'Health Status' section is highlighted with a green background, displaying the BMI value and the status 'Normal'. A red measuring tape and a round analog scale are visible on the right side of the form.

Form1

Weight: 85 KG
Height: 1,77 M

Calculate

Health Status:

Your BMI is: 27,1314117909924
Status: Overweight

This screenshot shows the BMI calculator interface with a pink background. The weight is 85 KG and height is 1,77 M. The 'Calculate' button is a light blue rectangle. Below it, the 'Health Status' section is highlighted with a yellow background, displaying the BMI value and the status 'Overweight'. A red measuring tape and a round analog scale are visible on the right side of the form.

Form1

Weight: 130 KG
Height: 1,77 M

Calculate

Health Status:

Your BMI is: 41,4951003862236
Status: Obese

This screenshot shows the BMI calculator interface with a pink background. The weight is 130 KG and height is 1,77 M. The 'Calculate' button is a light blue rectangle. Below it, the 'Health Status' section is highlighted with a red background, displaying the BMI value and the status 'Obese'. A red measuring tape and a round analog scale are visible on the right side of the form.

Practical Exercise 4b

Solution

```
1 reference  
private void btnCalculate_Click(object sender, EventArgs e)  
{  
    double bmi, weight, height;  
  
    weight = Convert.ToDouble(txtWeight.Text);  
    height = Convert.ToDouble(txtHeight.Text);  
  
    bmi = weight/ Math.Pow(height, 2);  
}
```

Next slide for more code



Practical Exercise 4b

Solution

```
if (bmi <= 18.4)
{
    rtbOutput.BackColor = Color.OrangeRed;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Underweight";
}
else if (bmi >= 18.5 && bmi <= 24.9)
{
    rtbOutput.BackColor = Color.Green;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Normal";
}
else if (bmi >= 25.0 && bmi <= 39.9)
{
    rtbOutput.BackColor = Color.Orange;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Overweight";
}
else if (bmi >= 40)
{
    rtbOutput.BackColor = Color.Red;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Obese";
}
```



Practical Exercise 4b

What about NESTED decision statements?



Practical Exercise 4c

Recommended Controls for both sections:

Section 1 (Login section)

- 3 labels
- 2 textboxes
- 2 buttons

Section 2 (Unit Calculation section)

- 2 labels
- 2 textboxes
- 1 button
- 1 richtextbox



Practical Exercise 4c

Login section

You will be tasked with the job of validating a user's login credentials before moving onto anything else.

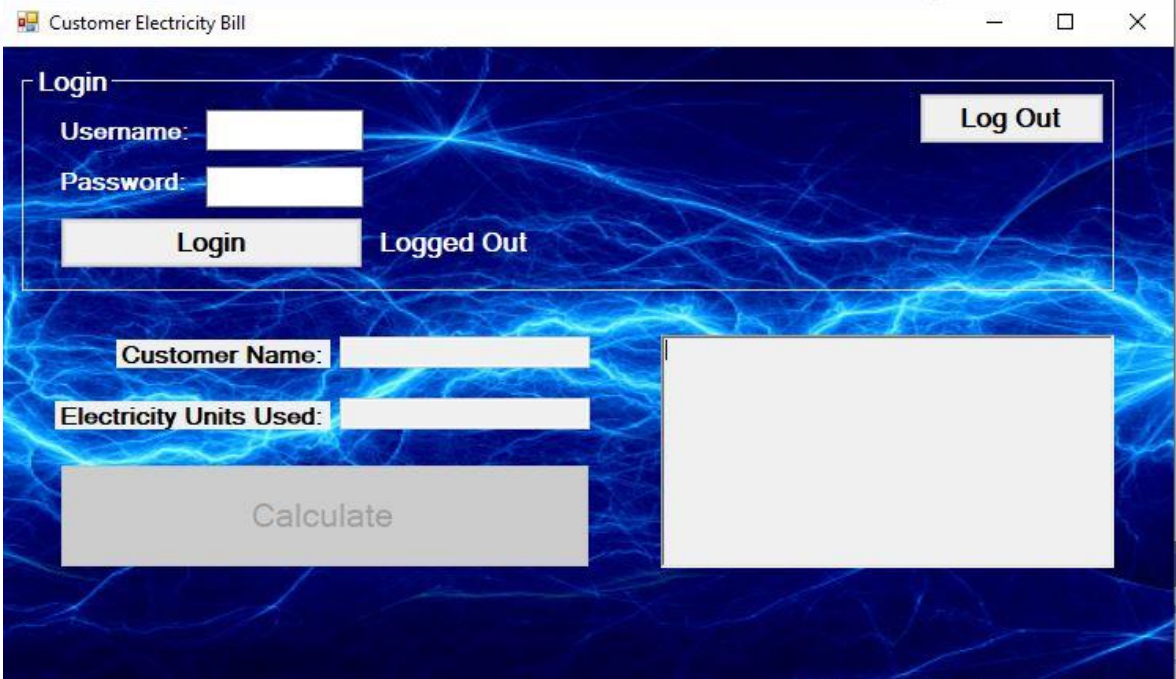
The username and password has already been given as follows:

Username = user

Password = 1234

Hint: Make these variables/constants

The Expected form design:



The screenshot shows a window titled "Customer Electricity Bill". Inside, there is a "Login" section with two input fields: "Username:" and "Password:". Below these is a "Login" button. To the right of the "Login" button is a "Logged Out" label. In the top right corner of the window is a "Log Out" button. Below the login section, there is a "Customer Name:" input field, an "Electricity Units Used:" input field, and a large "Calculate" button. To the right of the "Calculate" button is a large empty rectangular box, likely for displaying the result of the calculation. The background of the window has a blue, abstract, glowing pattern.

Practical Exercise 4c

Notice that the Calculate button (btnCalculate) as well as the textboxes (txtCustomerName and txtUnitsUsed) are disabled at default. This is because the user needs to first log in successfully before they can access and enter their details below in the unit calculation section.

The Expected form design:

The screenshot shows a Windows application window titled "Customer Electricity Bill". The window has a blue background with a lightning bolt pattern. It is divided into two main sections:

- Login Section:** Located at the top, it contains a "Username:" label, a text input field, a "Password:" label, another text input field, a "Login" button, and a "Logged Out" label. A "Log Out" button is also present in the top right corner of this section.
- Unit Calculation Section:** Located below the login section, it contains a "Customer Name:" label, a text input field, an "Electricity Units Used:" label, another text input field, and a large "Calculate" button. A large empty text area is to the right of the input fields.

Arrows from the text "The Expected form design:" point to these sections. A box labeled "Login Section" points to the top section, and a box labeled "Unit Calculation Section" points to the bottom section.

Hint: You will need to first validate the login credentials in order to enable the textboxes (txtCustomerName and txtUnitsUsed) and the calculate button (btnCalculate)

Practical Exercise 4c

Validation messages via messageBoxes: Communicate your validation outcomes with the user !

The Expected form design:

If username is incorrect

The screenshot shows a Windows application window titled "Customer Electricity Bill". The interface has a blue background with a network-like pattern. It contains a "Login" section with "Username:" and "Password:" labels, each followed by a text input field. The "Username" field contains the text "incorrect". Below these fields are "Login" and "Logged Out" buttons. To the right of the "Login" section is a "Log Out" button. Below the login section are "Customer Name:" and "Electricity Units Used:" labels, each followed by a text input field. At the bottom is a large "Calculate" button. A modal message box is displayed in the center-right, with the text "Please enter the correct Username" and an "OK" button.

If username is correct, but the password is incorrect

The screenshot shows the same "Customer Electricity Bill" application window. In this state, the "Username" field contains the text "user" and the "Password" field contains the text "0000". The "Log Out" button is now disabled. A modal message box is displayed in the center-right, with the text "Incorrect Password! Please enter the correct password." and an "OK" button.



Practical Exercise 4c

Validation visuals via messageBoxes. Communicate your validations with the user.

The Expected form design:

If the
username
and password
both match

The screenshot shows a Windows application window titled "Customer Electricity Bill". The window has a dark blue background with a glowing blue network pattern. The "Login" section contains a "Username:" field with the text "user" and a "Password:" field with the text "1234". Below these fields are two buttons: "Login" and "Logged Out". A "Log Out" button is located in the top right corner. Below the login section, there are two input fields: "Customer Name:" and "Electricity Units Used:". A large grey "Calculate" button is at the bottom. A modal message box is displayed in the center, containing the text "You have been successfully logged in!" and an "OK" button. An arrow points from the text box on the left to the "Login" button.

All the necessary controls in the Unit Calculation section should now also be enabled.

Practical Exercise 4c

Unit Calculation section

Once you have successfully logged in, you will now need to calculate and display a customer's electricity bill. The Customer's Name, and Unit used should be displayed in a richtextbox along with the total bill amount that the customer must pay at the end of the month.

Charge rate for Units Used are as follows:

Units Used per Month	Charge Rate per Units Used
0 – 200	@ R2.50
201 – 400	@ R5.00
401 – 600	@ R7.50
601 and above	@ R10.00

If the total monthly bill exceeds R3000.00 then a supercharge of 12% will be added on.

For every customer, the minimum monthly bill should be R100.00.

Practical Exercise 4c

Expected Output: When bill is above R3000.00 and super charge is applied

Customer Electricity Bill

Login

Username:

Password:

Customer Name:

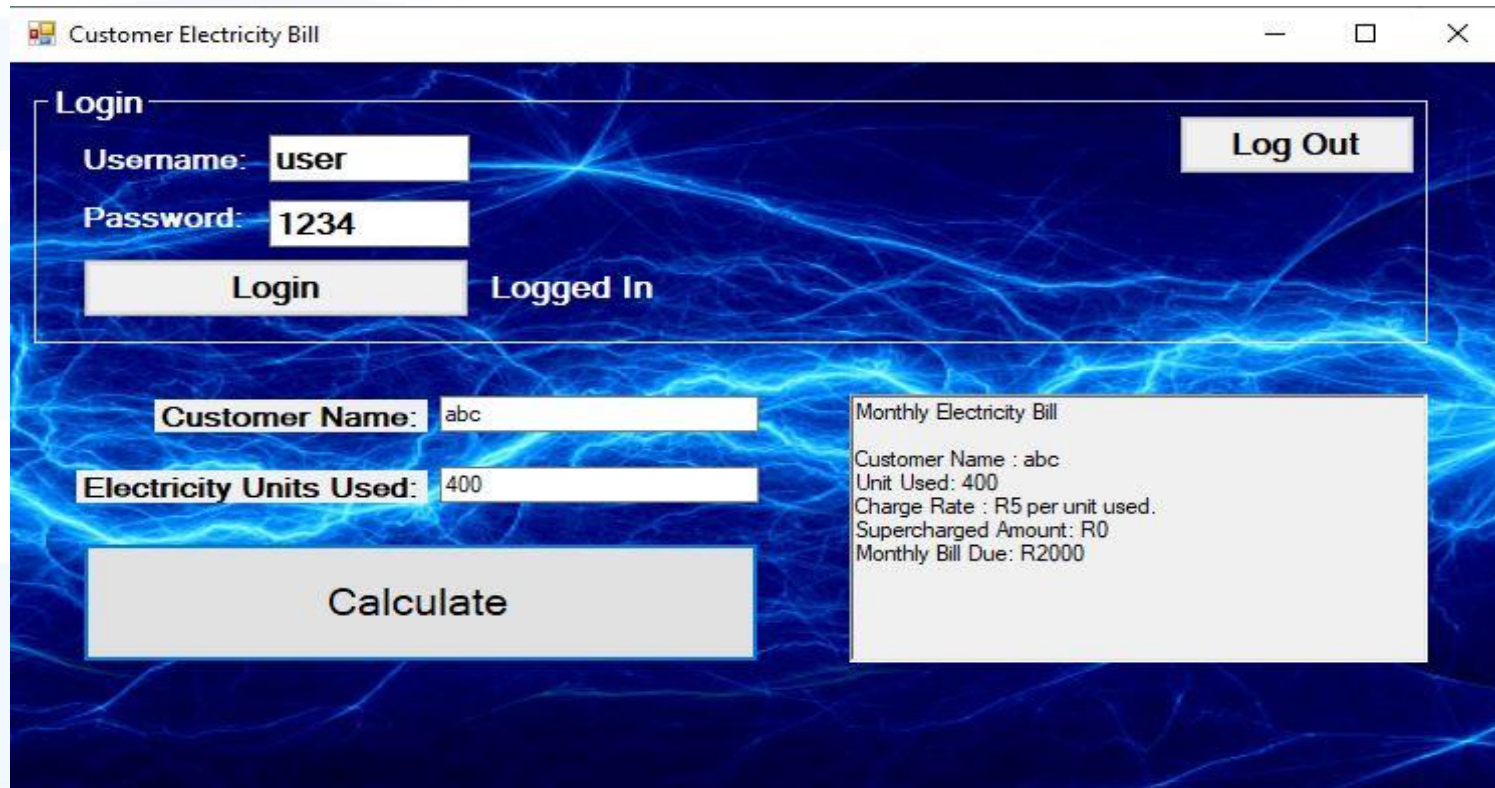
Electricity Units Used:

Monthly Electricity Bill

Customer Name : abc
Unit Used: 600
Charge Rate : R7,5 per unit used.
Supercharged Amount: R540
Monthly Bill Due: R5040

Practical Exercise 4c

Expected Output: When bill is below R3000.00 no super charge is applied



Customer Electricity Bill

Login

Username: user

Password: 1234

Login

Logged In

Log Out

Customer Name: abc

Electricity Units Used: 400

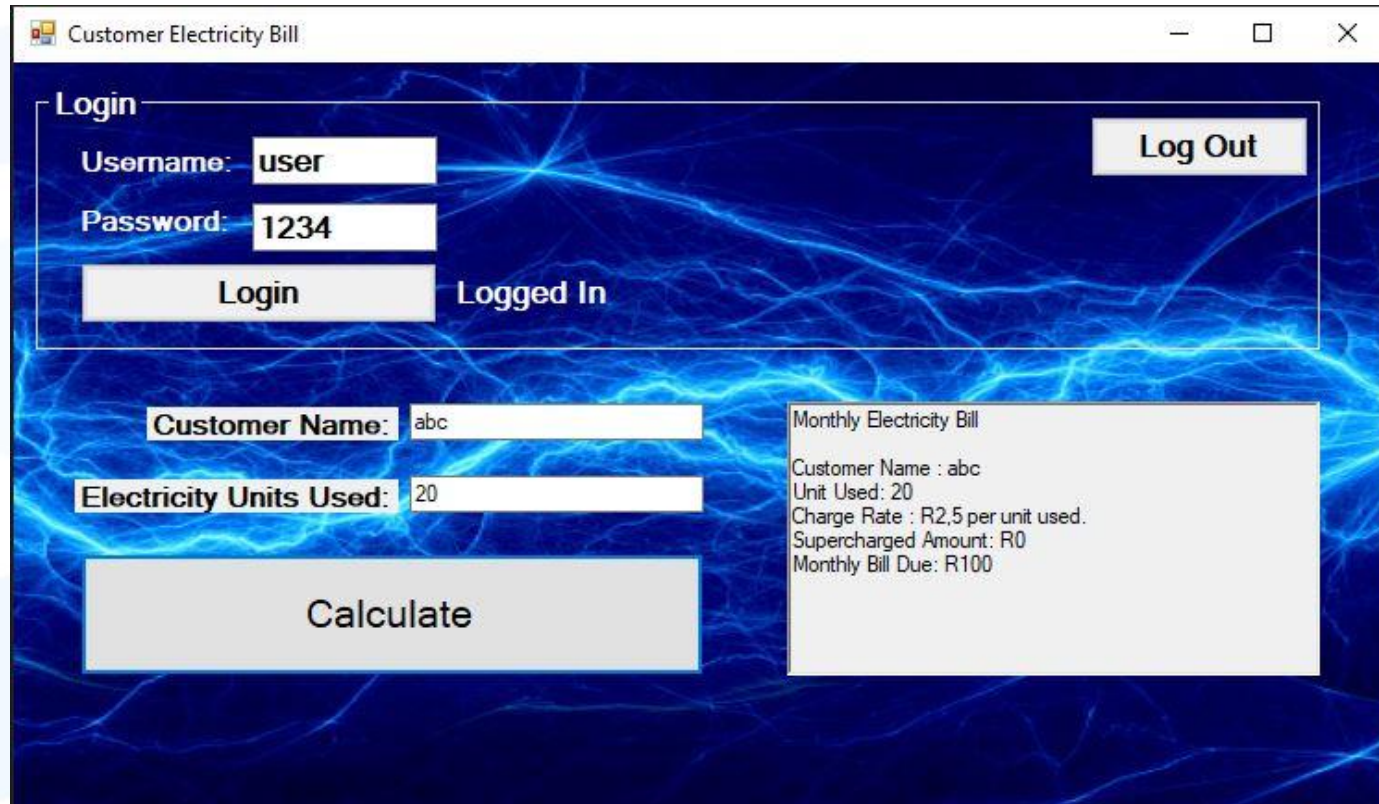
Calculate

Monthly Electricity Bill

Customer Name : abc
Unit Used : 400
Charge Rate : R5 per unit used.
Supercharged Amount: R0
Monthly Bill Due: R2000

Practical Exercise 4c

Expected Output: Minimum possible bill will always be R100.00



Customer Electricity Bill

Login

Username:

Password:

Logged In

Customer Name:

Electricity Units Used:

Monthly Electricity Bill

Customer Name : abc
Unit Used: 20
Charge Rate : R2,5 per unit used.
Supercharged Amount: R0
Monthly Bill Due: R100

Practical 4 Submission

Submit your Practical 3c project on ClickUP as follows:

- Due Date: 03 April 2023.**
- Name your project, **INF154Practical4xxxxxxxx** (Where xxxxxxxx is your student number) and compress (zip folder) your project.
- Submit it under the Practical 4 submission link.

