Cellphone Bill Word Problem

Python

Make sure the students have downloaded the Thonny program from Self-Service.

The Problem

Your cellphone plan costs $0.01 for every minute of talking, $0.005 for every text message, and $10 for each gigabyte of data. In addition, you pay $5.32 in federal, state, and local taxes. Write a program that calculates your monthly cell phone bill.

Given Information

The problem tells us how much the plan charges for talk, tech, and data. In addition, they tell us how much we pay in taxes. Create variables for each of these pieces of information.

```
talk_rate = 0.01
text_rate = 0.005
data_rate = 10
taxes = 5.32
```

Getting Information from the User

To solve this problem, we need to know how many minutes, how many texts, and how much data is used each month. We want our program to ask the user a question. To do this, we are going create a variable called `talk`, but we are going to use the `input` command to write a message to the user. To make sure this is working properly, we are going to write this value to the shell with the `print` command.

```
talk = input("Enter the number of minutes: ")
print(talk)
```

Adding This Information Together

Let's add another variable `text` that prompts the user for how many text messages they sent. Instead of

printing `talk` to the shell, lets add `talk` and `text` together and print this to the shell. What happens if you enter 5 for `talk` and 10 for `text`.

```
talk = input("Enter the number of minutes: ")
text = input("Enter the number of text messages: ")
print(talk + text)
```

Does this work? Why do we get `510` instead of `15`? When you use the `input` command, the computer stores this value as a string. A string is another way of saying text. So we got "five" the word instead of "five" the number. Python can only add numbers together. When Python adds two strings together, we call this concatination. This just smooshes the two values of `talk` and `text` together, hence `510`.

String to Number

To properly do addition, we need to turn a string into a number. We do this with the `int` command. `int` takes a string and returns an integer. Why is this a problem? While we don't send fractions of a text message, it is likely that you have used fractions of a minute of talking or part of a gigabyte of data. So, we don't want use `int` because integers are whole numbers. We want decimal points. Instead, we are going to use `float`.

```
talk = float(input("Enter the number of minutes: "))
text = float(input("Enter the number of text messages: "))
print(talk + text)
```

Now, you should be able to enter two numbers and have Python add them together. Erase the `print` command and add a line of code that prompts the user for their data usage.

Calculating Each Part

Our cellphone bill is divided into three parts - talk, text, and data. In order to calculate our total bill, we need to know how much each part of the bill costs. To do this, we are going to multiply the quantity by its rate. We are going to create three new variables `talk_amount`, `text_amount`, and `data_amount`.

```
talk_amount = talk * talk_rate
text_amount = text * text_rate
data_amount = data * data_rate
```

Finishing the Program

The next step is to add up all the components of our bill; don't forget the taxes. We will do this by using the

variable `total`.

```
total = talk_amount + text_amount + data_amount + taxes
```

The last step is to print the cellphone bill to the shell. Be sure to remember that all messages to the user should be a complete thought, i.e. use complete sentences, capitalization, proper spelling, etc.

```
print("Your montly cellphone bill is $" + total + ".")
```

If you run this program, you will notice that you get an error message. The message you print to the shell is one big string. However, we add the value from the variable `total` into the middle of the string. Python does not like having a number inside a string. We need to convert the value of `total` to a string so it can be printed to the shell.

```
print("Your montly cellphone bill is $" + str(total) + ".")
```