Answers to Questions from TT1.2

Name:
Student ID:

1. Desk Check Task: Calculate Bill Total

Required Variables:

Real (floating point):

appetizer_price, main_price, dessert_price

total_price

Pseudocode:

Read the value of appetizer_price

Read the value of main_price

Read the value of dessert_price

total_price = appetizer_price + main_price + dessert_price

Print '$' then the value of total_price *to the terminal showing two decimal places.*

Test Data:

	First data set	Second data set
appetizer_price	10.30	12.40
main_price	34.00	41.00
dessert_price	8.50	9.80

Expected Result:

	First data set	Second data set
Output:	$52.80	$63.20

Desk check - fill this in by hand-tracing/hand-executing the pseudocode provided with the test data above:

	Statement	*appetizer_price*	*main_price*	*dessert_price*	*total_price*	*output*
First Pass	*Read the value of appetizer_price*	10.30	—	—	—	—
	Read the value of main_price	—	34.00	—	—	—
	Read the value of dessert_price	—	—	8.50	—	—
	Calculate the total_price	—	—	—	52.80	—
	Convert to dollars	—	—	—	$	—
	Output the total_price	—	—	—	—	$52.80
Second Pass	*Read the value of appetizer_price*	12.40	—	—	—	—
	Read the value of main_price	—	41.00	—	—	—
	Read the value of dessert_price	—	—	9.80	—	—
	Calculate the total_price	—	—	—	63.20	—
	Convert to dollars	—	—	—	$	—
	Output the total_price	—	—	—	—	$63.20

2. Complete Program Calculate Bill Total

Now check the actual code produces the output you expected

Do this by completing the missing code in **bill_total.rb** in **Task 1.3** then running the program.

3. Short Answer Questions:

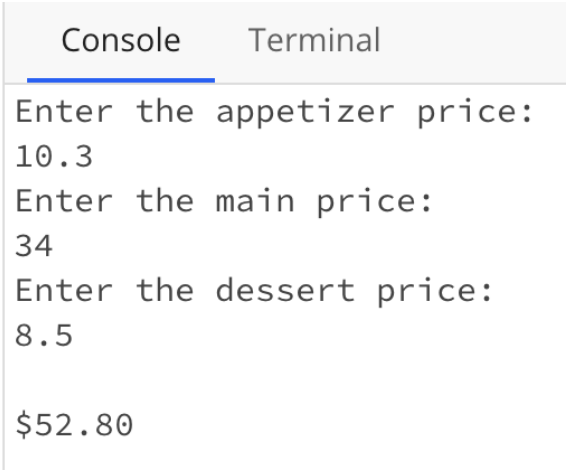
Focus in the following on using the correct computing terminology.

Here are some terms that may help you: Assignment, evaluate, increment,

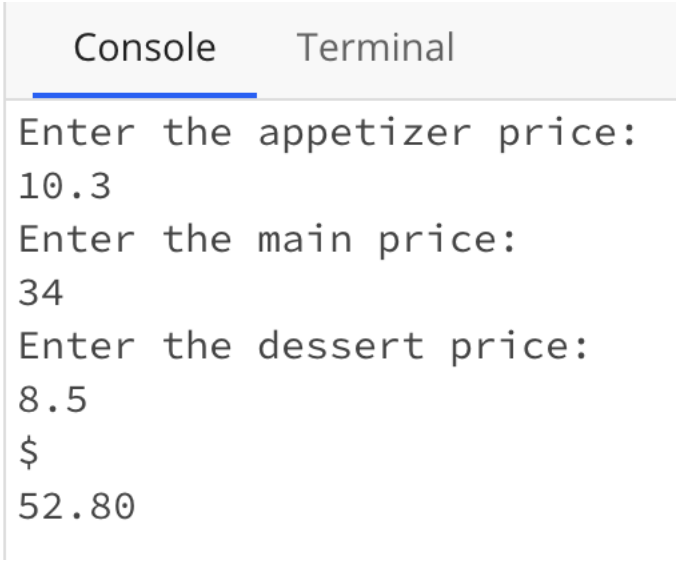
1. Using a few sentences explain why it may be important to execute statements in the correct sequence. (eg: what might happen if the last statement in Program 2 was executed earlier)

   It is considered good practice and good habits to write the code in the order that the output is desired. However in this code line 21 isn't required because the previous output line was line 14, which was a Puts string, a line break is automatically added.

   Putting line 21 above instead like in line 19 would result in in the output having an extra line break.



   Putting line 21 in line 20 however would result in the dollar sign being printed first,



2: The code main_price = 10 is an example of which kind of programming statement?

   This is an assignment statement. The variable **main_price** is assigned a value of **10**

3: What **actions** does the computer perform when it executes `a = a + b`?

The computer first evaluates the sum of variables **a** and **b**
Then it assigns the new sum as the value of variables **a**

4: How would the value of variable i change in the statement `i = i + 1`?

The value of i will be increased by an increment of 1.

5: *What sort of types will Ruby use to store the following variables* (given the associated variable values)?

Data	Type
A person's name e.g: “Fred Smith”	String
Number of students in a class e.g: 23	Integer
Average age of a group of people e.g: 23.5	Float
A temperature in Celsius e.g: 45.7	Float
True or false e.g: 1 == 2	Boolean

Note: possible types include: Integer, String, Float, Boolean

6: *Variables have a scope – what are two different scopes variables can have in Ruby?*

One scope is the global variable, it is not recommended to do so as with large codes it will be hard to read. The value of the variable is set for the whole code.

Another scope is the class variable. This variable is defined within the class and shared among its instances

See the lesson materials for help with Question 6. You could also see:

https://www.tutorialspoint.com/ruby/ruby_variables.htm