

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
<i>appetizer_price</i>	10.30	12.40
<i>main_price</i>	34.00	41.00
<i>dessert_price</i>	8.50	9.80

Expected Result:

	First data set	Second data set
<i>Output:</i>	\$52.80	\$63.20

Desk check - fill this in by completing the missing code in **bill_total.rb** (in the tasks Resources folder) then running it with the test data above:

	Statement	<i>appetizer _price</i>	<i>main _price</i>	<i>dessert _price</i>	<i>total _price</i>	<i>output</i>
First Pass	<i>Read the value of appetizer_price</i>	10.30	0	0	0	0
	<i>Read the value of main_price</i>	10.30	34.00	0	0	0
	<i>Read the value of dessert_price</i>	10.30	34.00	8.50	0	0
	<i>Calculate the total_price</i>	10.30	34.00	8.50	52.80	0
	<i>Convert to dollars</i>	10.30	34.00	8.50	52.80	\$
	<i>Output the total_price</i>	10.30	34.00	8.50	52.80	\$52.80
Second Pass	<i>Read the value of appetizer_price</i>	12.40	0	0	0	0
	<i>Read the value of main_price</i>	12.40	41.00	0	0	0
	<i>Read the value of dessert_price</i>	12.40	41.00	9.80	0	0
	<i>Calculate the total_price</i>	12.40	41.00	9.80	63.20	0
	<i>Convert to dollars</i>	12.40	41.00	9.80	63.20	\$
	<i>Output the total_price</i>	12.40	41.00	9.80	63.20	\$63.20

2. 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 crucial that steps in an algorithm are performed in the right order, otherwise the algorithm might not work correctly. For example:

```
print("$")
printf("%.2f", total_price)
```

⇒ The output would be "\$" + "total_price" (\$52.80).

However, if the last statement was executed earlier:

```
printf("%.2f", total_price)
print("$")
```

⇒ The output would be "total_price" + "\$" (52.80\$) and this would not match the expected result.

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

This is an assignment statement, specifically, assign values from right side operands to left side operands.

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

The computer first assigns a to the value of a + b.
Then it calculates a + b.

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

The value of i will be increased by 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?

Variables can have global scope or local scope. Global scope means that the variable is accessible to all blocks of code; however, variable with local scope can only be accessible within the block which it is first used (or defined).

Take the following blocks of code for example:

```
$fRACTION = 0.5
def add_two_numbers(a, b)
  total = a + b
  puts 'total is: ' + total.to_s
end

def main()
  puts total.to_s
  puts $fRACTION.to_s
  add_two_numbers(16, 5)
end

main()
```

- ⇒ In this example, the variable `$fRACTION` has global scope and it is accessible anywhere in the program; however, the variable `total` has local scope and it is only accessible with the function `add_two_numbers`, therefore, calling it anywhere else will result in an error.

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

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