Faculty of Science, Engineering and Technology

Introduction to Programming

Pass Task 2.3: Hospital Charges

Overview

Procedures are a useful tool for capturing the instructions needed to perform a task, but sometimes you need to be able to capture the instructions needed to calculate a value. Using functions you can now create artefacts to encapsulate the steps needed to calculate a value.

Purpose:	Learn how to use functions and write your own functions.
Task:	Write and use some simple functions and procedures
Time:	This task should be completed before the start of week 3.
Resources:	▪ Chapters 6 and 8 of 'Learn to Program' **Chapt 6: Flow Control** **Chapt 8: Writing Your Own Methods** ▪ Learn Ruby the Hard Way: **Learn Ruby the Hard Way: Functions can return something.**

Submission Details

You must submit the following files to Canvas:

- Program source code demonstrating your creation and use of functions
- Screenshot of the Terminal showing the execution of your program.

Make sure that your task has the following in your submission:

- The completed functions and procedures based on the instructions here and in the code.
- Code must follow the Ruby coding convention used in the unit (layout, and use of case).
- The code must run and the screenshot show it working.

SWINBURNE UNIVERSITY OF TECHNOLOGY

Instructions

Functions allow you to build modular programs. Functions take arguments and return a result.

To explore this topic, we will complete a program that calculates hospital charges so that it will:

- Prompt the user to enter a patient name
- Call functions to sum the charges that make up the total amount owed by the patient.
- Print to the terminal the patient's name and total owed.
- Use the functions provided in the file **input_functions.rb** to read values and print the total.
- Compile and run your program. Below is a sample of the program running:

Steps in the program:

1. Read in the patient's name (you need to write this – use the *read_string(prompt)* function from the file **input_fumctions.rb.**
   NOTE: you do not need to copy in the functions from input_functions.rb – they are available because of the 'require './input_functions.rb' line at the top of the program.
2. Calculate the accommodation charges (these are just read in)
3. Calculate the theatre charges (these are just read in)
4. Read in the pathology charges (you need to write this to read the in)
5. Print to the terminal the patient's name and the bill total (using the function *print_float(value, decimal_places)* from the file **input functions.rb.**

The output of the program should look as follows:

```
Enter patient name:
Sam Jones
Enter the accommodation charges:
23.50
Enter the theatre charges:
45.99
Enter the pathology charges:
59.20
The patient name: Sam Jones
The total amount due is: $128.69
```

Once the Task is complete you can submit it for assessment, which will help prepare it for your portfolio.

1. Use **Skitch** (or your preferred screenshot program) to take a screenshot of the Terminal, as this is one of the things you will need to submit.
2. Save the document and backup your work to multiple locations!
   - Once you get things working you **do not** want to lose them.
   - Work on your computer's storage device most of the time... but backup your work when you finish each task.
   - Use **Dropbox** or a similar online storage provider, as well as other locations.
   - Canvas is not a Backup of your work, so make sure you keep a copy!
   - A USB key and portable hard drives are good secondary backups... but can be lost/ damaged (do not rely upon them).
3. Login to Canvas, and locate Pass Task 2.2
4. Change the status of the task to **Ready To Mark**
5. Upload your completed code and the screenshot.
6. If you check back later Canvas will have prepared these as PDFs for your tutor to assess.

You now have another of your first portfolio pieces. This will help demonstrate your learning from the unit.

Note: This is one of the tasks you need to **submit to Canvas**. Check the assessment criteria for the important aspects your tutor will mark.

End of Task