Faculty of Science, Engineering and Technology

Introduction to Programming

Pass Task 1.3: Reading and Writing (Hello User)

Overview

This program will allow you to demonstrate simple reading and writing from the terminal window and using basic data types in Ruby.

Purpose:	Develop a simple interactive Ruby program working with basic data types.
Task:	Create your own Reading and Writing program using the Ruby language. Submit to Canvas when complete.
Time:	This task should be started in your first lab class and submitted for feedback before the end of week 2.
Resources:	**Pine, C 2013 *Learn to Program*, The Pragmatic Programmer. (available in the library).** **Learning Material for Week 1.**

Submission Details

You must submit the following files to Canvas:

- Hello User source code (hello_user.rb)
- Screenshot of the Terminal showing the execution of your Hello User program.

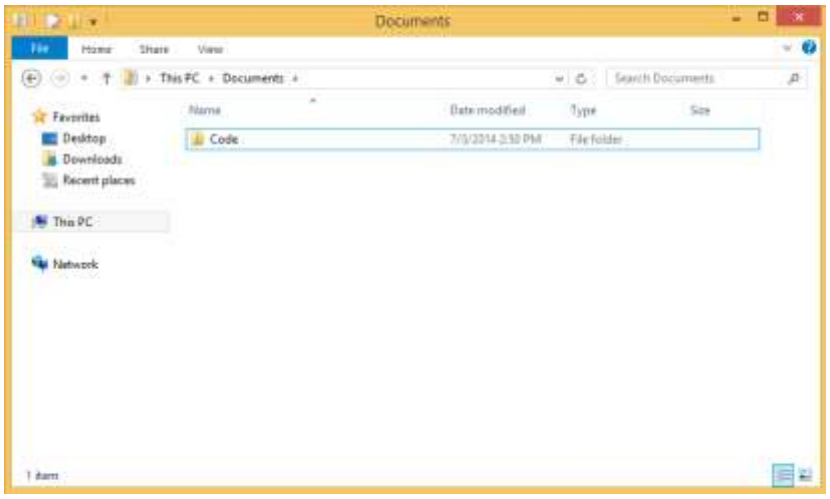
Make sure that your task has the following in your submission:

- The program must read the required details from the user, and respond correctly
- 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.
- Your program must have a procedure for Main.
- Your program must have the indicated local variables, and use them appropriately.

Preparation.

If you have not already done it then you will need to follow the steps to install the tools you will need to use Ruby in this unit

1. If using a laptop Install the tools you need to get started for your operating system. Install **Atom**, **SublimeText or Notepad++ or equivalent** (or use a text editor of your choice) see the “install Ruby” notes on Blackboard/Canvas
2. If you don’t already have one, make a directory (i.e., a ‘folder’) to store your code (e.g., *Documents/Code/Lab1*). On a Swinburne computer you may wish to use a directory on your student drive or a USB storage device.
   - Navigate to your *Documents* directory in Finder or File Explorer
   - Right click in the *Documents* directory and select **New Folder**, name it **Code**



Note: You can skip this step on the computers in the Swinburne lab as this will already be setup.

Instructions

Variables allow you to store values that can change in your program. When you declare a local variable you are telling the computer to create a variable for use within the function or procedure. You must give the variable a name and keep in mind the type of data it will store. The computer will then set aside space for you to store a value for that variable. You can then write values to the variable and read values back.

To explore this topic, we will create a Terminal program that will:

- ask the user to enter their name, age, and weight in centimetres,
- calculate and output the year the user was born, and
- calculate and output the user's height in inches

1. Declare the Hello User program using the program start code shown below (available in the Resources for the task.

```
require 'date'

INCHES = 39.3701  # This is a global constant

# Insert the missing code here into the statements below:
# gets
# gets.chomp
# Date.today.year
# year_born.to_i
# gets.to_f

def main
   puts 'What is your name?'
   name =
   puts 'Your name is ' + name + '!'
   puts 'What is your family name?'
   family_name =
   puts 'Your family name is: ' + family_name + '!'
   puts 'What year were you born?'
   year_born =
   # Calculate the users age
   age =
   puts 'So you are ' + age.to_s + ' years old'
   puts 'Enter your height in metres (i.e as a float): '
   value =  # Should this be a float or an Integer? Why?
   value = value * INCHES
   puts 'Your height in inches is: '
   puts value.to_s
   puts 'Finished'
end

main  # call the main procedure
```

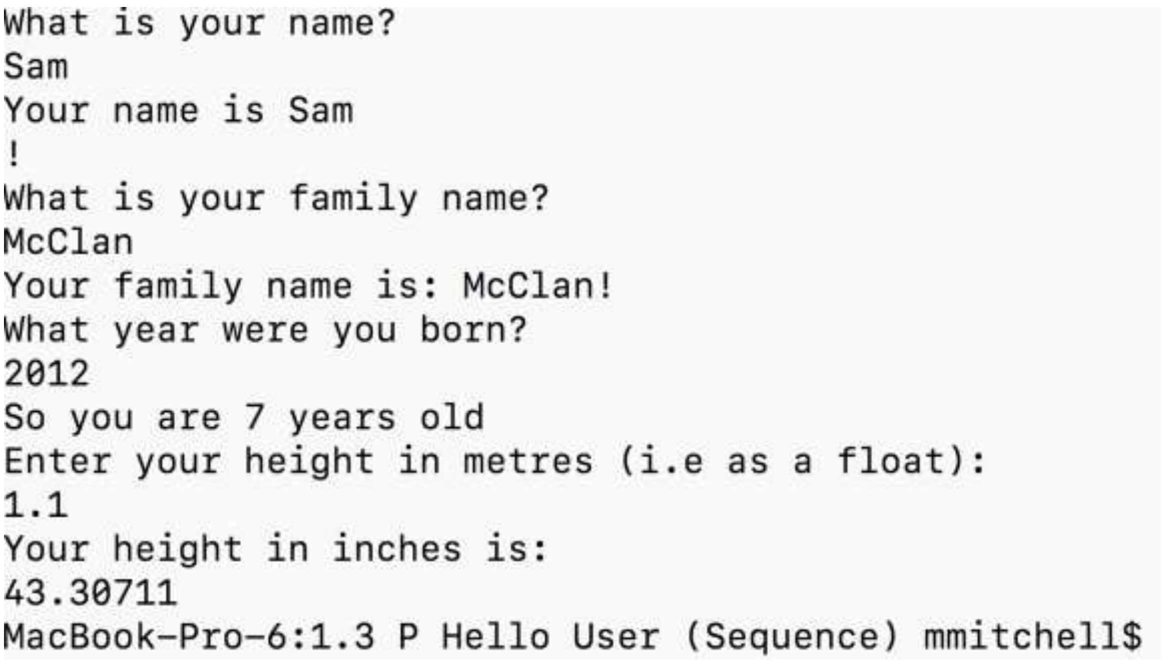
Tip: This code is the basic program template that you can use whenever you create a new program. The *Main* procedure will have the program's main instructions.

2. Extend the code in main so that it does the following:

```
1. Read the user's name (a String, prompt with 'Please enter
   your name: ') and store it in the name variable.
2. Print out the user’s name followed by an exclamation mark.
3. Read the user's family name, remove any white space and
   store it in the family_name variable.
4. Print out the user’s family_name followed by an exclamation
   mark.
5. Read in the user’s year of birth
6. Convert the year of birth to an integer then calculate the
   user’s age and print it out.
7. Prompt for and read in the user’s height in metres
8. Convert the height to inches and print out the result.
```

Note: Don't forget to use under scores when declaring variable names containing more than one word (e.g., use year_born (**not** yearborn or YearBorn)).

An example of the output of the program is as follows:



```
What is your name?
Sam
Your name is Sam
!
What is your family name?
McClan
Your family name is: McClan!
What year were you born?
2012
So you are 7 years old
Enter your height in metres (i.e as a float):
1.1
Your height in inches is:
43.30711
MacBook-Pro-6:1.3 P Hello User (Sequence) mmitchell$
```

Now that 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 1.3
4. Change the status of the task to **Ready To Mark**
5. Upload your completed Hello User 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 aspect your tutor will check.