Faculty of Science, Engineering and Technology

Introduction to Programming

Tutorial Task 4.1: File Handling

Overview

This program will allow you to demonstrate simple reading and writing from a file using Ruby

Purpose:	Modify a simple Ruby program that reads and writes from a file to use loops
Task:	Create your own file reading
Submit To:	Canvas when complete
Time:	This task should be started in your fourth lab class.
Resources:	**Tutorials Point**
	File class (Ruby documentation)
	IO class (Ruby documentation)

Submission Details

You must submit the following files to Canvas:

- Basic read-write source code (*basic_read_write.rb*)
- Screenshot of the Terminal showing the execution of your Basic read-write program.

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

- The program must read the required details from the file, and then print out each line to the terminal.
- 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.

Instructions

Files allow you to store data persistently. In this task you will write a simple file reading program to read multiple lines using a loop printing each line read to the terminal screen.

To explore this topic, we will modify a Terminal program that will, when complete:

- Open a file and write a number of records to the file. Close the file.
- Open a file and loop according to the number of records to read in each record.
- Print each record that is read (as they are read).

Use the code provided (from this task's resources in Canvas) to get started, using this code complete the following:

1. Open and look at the code in the Resources for the basic code for reading and writing the records from files. The functionality of this code is basically correct, but the code can be improved in design and implementation, these are the modifications you will make.
2. Make the following modifications the *basic_read_write.rb* program so that it:
    - Uses a loop in *read_data_from_file()*, with the loop controlled by the number at the start of the file.
    - Improve the functional decomposition by removing as many lines of code from main as possible, yet retaining good structure.

The program output should look as follows:

```
MacBook-Pro-6:4.1T File Handling mmitchell$ ruby basic_read_write_answer.rb
Fred
Sam
Jill
Jenny
Zorro
MacBook-Pro-6:4.1T File Handling mmitchell$
```

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

1. Use your preferred screenshot program to take a screenshot of the Terminal, as this is one of the things you will need to submit.
2. 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 Tutorial Task 4.1
4. Change the status of the task to **Ready To Mark**
5. Upload your completed File Handling 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**. Tutors should give guidance and perhaps feedback in the tutorial class.
Check the assessment criteria for the important aspect your tutor MAY check when assessing your finished portfolio.

End of Task