



# Introduction to Programming

## Tutorial Task 3.2: Simple Menu

### Overview

In this task you combine control flow with modularization to create the basis of a menu system for a text-based (terminal based) Music player.

**Purpose:** Learn to combine control flow and modularisation.

**Task:** Create a program that implements a basic menu system.

**Time:** This task should be completed before the start of week 6.

**Resources:**

- [Using a case statement in Ruby](#)
- Chapters 6 and 8 of 'Learn to Program'
  - [Chapt 6: Flow Control](#)
  - [Chapt 8: Writing Your Own Methods](#)

**Note:** Remember to submit **all tasks** to Doubtfire for assessment. Also make sure you *fix and resubmit* any tasks you did not get signed off last week!

### Submission Details

You must submit the following files to Doubtfire:

- SimpleMenu program source code.
- A screen shot of the program running

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

- Demonstrates use of Ruby programming convention, including indentation within selection and repetition statements.
- Demonstrates use of a **case** statement to perform selection, and a **repeat until** loop to perform repetition.

## Instructions

Modify a small program that will give the user the following options:

1. Enter or Update an album
  2. Play an existing album
  3. Exit the system
1. Download and extract the resources for this task.
  2. Open **simple\_menu.rb** using Atom (or similar).
  3. Implement a **maintain\_albums** procedure with the following logic:
    - A repeat-until loop with the following:
      - Display the sub-menu options and prompt the user to select one
      - Read in the user selection and then use a case statement to call a procedure that corresponds to the selection (or leave the loop if they chose exit)
      - The called procedure should display a message to user about the option. You do not need to implement the procedure at this stage, just leave it as a 'stub' that displays a message.

Create a stub for main menu option 1.

Steps:

- 1: Displays the sub menu options:  

```
'Sub Menu Maintain Albums: Enter your selection:'  
'1 To Enter Album'  
'2 To Update existing Album'  
'3 Return to Main Menu'
```
- 2: For each sub-menu option create a stub.
- 3: Each stub should wait until the user presses enter, then return to the sub menu.

4. Now implement a stub for the second main menu option so that main procedure calls the stub. Call the stub *play\_existing\_album*. See the box below:

Create a stub for main menu option 2.

Steps:

1: Displays the menu options:

```
'Main Menu: Enter your selection:'  
'1 To Enter or Update Album'  
'2 To Play existing Album'  
'3 Exit'
```

2: The stub should display "You selected Play Album.  
Press enter to continue"

3: The stub should wait until the user presses enter,  
then returns to the main menu.

Upload to Doubtfire:

1. Your *simple\_menu.rb* code
2. A screenshot of your code running

End of Task