Faculty of Science, Engineering and Technology

Introduction to Programming

Tutorial Task 2.1: Using Functions (Hello User with Functions)

Overview

In this task we change our Reading and Writing (Hello User) program to use functions to prompt and read in values.

Purpose: Learn how to use functions.

Task: Use the instructions on the following pages to **modify** your own Hello User program from Pass Task 1.1. Submit to Doubtfire when complete.

Time: This task should be completed in your lab class and submitted for feedback before the start of week 4.

Resources:

- Chapter 4 of the Programming Arcana
- Swinburne CodeCasts (YouTube Channel, iTunesU)
    - Variables
- Syntax Videos
    - Assigning Values to Variables, Declaring Local Variables, Calling Functions

Submission Details

You must submit the following files to Doubtfire:

- Modified Reading and Writing (Hello User) source code (HelloUserWithFunctions.pas)
- Screenshot of the Terminal showing the execution of your modified Hello User program.

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

- The program must use the TerminalInputFunctions to read the required details from the user, and respond correctly.
- Code must follow the Pascal coding convention used in the unit (layout, and use of case).
- The code must compile 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.

SWINBURNE UNIVERSITY OF TECHNOLOGY

Instructions

Procedures allow you to ‘chunk’ together related lines of code. Procedures can receive data as arguments, but procedures do not allow you to return a value. Functions allow both data to be passed in and a value returned.

The functions we are going to use in this task are contained in the file UserInputFunctions.pas (see the resources for this task). We are going to modify our previous program to use these functions.

If you recall our Reading and Writing (Hello User) program did the following:

- ask the user to enter their name, age, and the current year,
- calculate the year the user was born, and
- output a greeting message.

1. Download the **Terminal User Input** pascal code from Blackboard. This unit contains some useful functions you can use to read values from the user.
2. Copy TerminalUserInput.pas into the folder next to your HelloUser.pas file.
3. Using Sublime Text declare the HelloUserWithFunctions program based on the program start code shown below.

```
program HelloUserWithFunctions;
uses TerminalUserInput;

procedure Main();
var
    name: String;
    age: Integer;
begin
    name := ReadString('Please enter your name: ');
    age := ReadInteger('How old are you this year? : ');
end;

begin
    Main();
end.
```

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

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

```
Procedure: Main
---------------------------------
Variables:
 - name (to store a String value)
 - age  (to store an Integer value)
 - year (to store an Integer value)
 - yearBorn (to store an Integer value)
---------------------------------
Steps:
 1: Read the user's name (a String, prompt with 'Please enter your name: ') and store it in the name variable.
 2: Read the user's age (an Integer, prompt with 'How old are you this year? : ') and store it in the age variable.
 3: Read the year (an Integer, prompt with 'What year is it? : ') and store it in the year variable.
 4: Calculate the year the user was born (year - age), and store it in the yearBorn variable
 8: Output the message, 'Hello ', name, ', you were born in ', then yearBorn
 9: Output the message “Press enter to continue”
 10: Read in a blank line.
```

5. Use Mingw with fpc to compile and run your program.

Hint: Remember the **WriteLn** procedure can be used to output messages.

Note: Don’t forget to use camel case when declaring variable names containing more than one word (e.g., use yearBorn (**not** yearborn or YearBorn)).

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.
   - Doubtfire 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 Doubtfire, and locate Tutorial Task 2.1
4. Change the status of the task to Ready To Mark
5. Upload your completed Hello World code and the screenshot.
6. If you check back later Doubtfire will have prepared these as PDFs for your tutor to assess.