Indirect Access

Now that we have nicely organised data and code, the next thing to look at is the idea of indirectly accessing data. This is a powerful concept that moves beyond directly accessing the data in a variable, to let you access the data indirectly through pointers and references.

Learning Goals

To complete this task, you need to demonstrate that you can do the following:

- Use pass-by reference to accept and update values.
- Use **const (constant)** references to provide read-only access to data to improve performance.
- Use pointers to refer to and interact with other values in memory.

Focus

- **Programming concepts:** Focus on how pointers and references allow you to interact with data elsewhere in memory. See how you can use this together with parameters and your structured data to update the entities within your program.
- **Programming process**: Focus on the way these new concepts give you the ability to create nice small functions and procedures to manipulate your entities, helping you better modularise your code.
- **Coding**: Focus on memorising the syntax for pointers and references. With pointers, focus on declaring pointers, getting the address of a value, and following a pointer to access a value.
- **Professional Characteristics:** Working with pointers requires extra attention to detail – now some variables can refer to other places. Keeping track of this is important to ensuring things work as expected.

Your Task

For this task you will need to submit the following:

- A PDF document containing:
  - Summaries and reflections.
  - Source code and a photo of the associated hand execution for your test program.
  - At least two (2) screenshots :
    - Your revised fly catch running.
    - Your test your knowledge program running.
  - Learning Journey and Resources
- Source code for:
  - The revised Fly Catch
  - Your chosen program from the Test Your Knowledge activities

1. Complete Learning Activities

Work through these steps to develop and demonstrate your understanding. Aim to demonstrate, to yourself and others, that you have achieved the learning goals.

1. Everything you need is in Chapter 4 Indirect Access from Part 2 of the Programmer's Field Guide.
2. Review details on the heap, pointers, and references.
3. Create a small program to demonstrate the use of pointers. Do *at least* the following in the program:
   a. Create:
      i. Two integer variables (v1 and v2)
      ii. A pointer to an integer (p)
   b. Initialise the integer variables.
   c. Output the values of the variables, and their addresses.
   d. Use p to update and print the value in v1:

        i. Assign p the address of the first variable (v1)
        ii. Use p to update and print the integer value
        iii. Print the address in the pointer, and the value from the variables to see the changes.
    e. Repeat with the second variable (v2) using the same pointer (p).
    f. In a loop that runs until the user wants to quit,
        i. Ask the user which variable they want to update
            1. Change the pointer to refer to that variable
        ii. Update and value via the pointer
        iii. Print the value of both variables

Show screenshots capturing your journey for building and running this program.

4. Use the hand execution process to demonstrate how this functions. **Capture** a photo of your hand execution to accompany your code.

5. Create a small program to demonstrate the use of pass-by reference. Do *at least* the following in the program:
    a. Create a **swap** procedure that will swap the value of two passed in parameters using pass-by reference.
    b. In **main**
        - Create three integer variables (v1, v2, v3)
        - Assign them unique values (eg 10, 20, 30)
        - Use your swap procedure to v1 and v2 – then print all three values.
        - Use your swap procedure to v2 and v3 – then print all three values.
        - Use your swap procedure to v1 and v3 – then print all three values.

    Show screenshots capturing your journey for building and running this program.

6. Use the hand execution process to demonstrate one of the calls to swap. **Capture** a photo of your hand execution to accompany your code.

7. Update your Fly Catch program to make use of pass by reference. **Capture** notes on your learning as you progress, indicating if and where you need to review the existing solutions.
8. Complete one of the Test Your Knowledge activities.
9. Prepare your summary, making sure to cover all learning goals and related concepts. Remember that this is a personal summary that demonstrates your understanding of the concepts.
10. Prepare your reflections by responding to the following:
    - What gives you confidence you have achieved the learning goals?
    - What is the most important thing you learned from this and why?
11. Capture your learning journey and collate your evidence of study and practice.

2. Upload Your Submission & Engage with Feedback

Mark the task as **Ready for Feedback** and upload the required files. Make sure to keep copies of these in case you need to resubmit. Then engage with the feedback you receive and get the task Complete!

If you are asked to resubmit, make sure your subsequent submission includes a comment that describes how you have addressed the feedback you received. This needs to demonstrate how you have addressed all the aspects indicated by your tutor in their feedback on your learning. If you don't understand the feedback, ask for clarification. If it is too generic, ask specific questions, only you know what feedback you need, take charge of it.