# Take Home Code Exercise

Hello and thanks so much for taking the time to do the Ferocia Code Exercise.

## Why are we asking you to do this?

*We want to talk to you about code - how you write it, how you think about it, how you turn ideas into concrete programs, what makes good code. We want to do this in a way that gives you the chance to show your skills at their finest, in a context that we both understand. This code exercise is designed to give us common context that we are both "experts" in so that we can have that talk.*

## What are we asking you to do?

We are a company that creates banking software. We'd like you to build a very simple term deposit calculator that takes as inputs:

- Start deposit amount (e.g. $10,000)
- Interest rate (e.g. 1.10%)
- Investment term (e.g. 3 years)
- Interest paid (monthly, quarterly, annually, at maturity)

And produces as output:

- Final balance (e.g. $10,330 on the above inputs, interest paid at maturity)

You can check your outputs using this calculator*. You can assume that all proceeds are reinvested into the term deposit for its duration.

**Non-functional requirements**

- CLI is fine, although of course if you want to make a simple UI because you think that best demonstrates your skills, please do. We won't give points to a beautiful UI so ask you to please spend your time on the code and not CSS if you do decide to create a UI.

- We want you to spend *at most* 2 hours on this as we need a conversation starter and not a fully pro application. Refer to the guide below for what we are looking for, and as long as your solution attempts to address each of these points you should consider yourself done. Please don't include setup (of computer/environment) time in the 2 hours, we all know how that can blow out.
- Any language is fine, and we recommend you choose your strongest language.
- Please submit using a link to a Github repository. If this could cause issues for you we will accept a .zip file with your code.

## What are we looking for?

- Usability - can we work out how to install and run it?
- Testing - have you written tests and do they pass?
- Functionality - does the program work as expected?
- Application design - is there clear separation of concerns?
- Simplicity - can we understand your code and how you've structured it?
- Feedback - how have you handled errors and let the user know?
- Code fluency - how proficient are you at the technologies you've chosen?

***HINT:** There are all kinds of compound interest equations out there. The ones that take year and month as arguments tend to be a lot more confusing than those that just take a single time input (e.g. month OR year, not both).