Loughborough University

COMPUTATIONAL PHYSICS 1 - PROBLEM SHEET: WEEK 1 (PHA904)

Try to answer this sheet without looking at your notes on your first attempt.
Note that you may find questions marked (*) challenging and those (**) very hard. This is meant to be the case - try, but do not be disappointed if you cannot succeed in solving these.
Last updated: Monday 5th October, 2020

---

1. Login and open Xcode, start a new Swift playground. Declare some variables (using `var`) and constants (using `let`). Do some basic arithmetic.

2. Declare some explicitly typed variables to double and integer precision. Find out what happens when you try to do arithmetical operations (`+,*,-,/`) with variables of a different type.

3. Define some simple functions such as the cube of a number (think about what functions you need to do some basic mathematics). Test your code.

(*) 4. Define a factorial function and test it – determine by trial and error the largest factorial you can compute. Can you increase this largest number?

5. Take the example code below and use it as the basis for forming your own `Complex struct` for representing a complex number (which has real and imaginary components):

```
public struct RationalNumber {
    public private(set) var numerator: Int
    public private(set) var denominator: Int

    public func times(_ other: RationalNumber )
    -> RationalNumber {
        return RationalNumber(
            numerator: numerator*other.numerator,
            denominator: denominator*other.denominator);
    }
```

```
    public func plus(_ other: RationalNumber )
    -> RationalNumber {
        return RationalNumber(
            numerator: numerator*other.denominator +
                    denominator*other.numerator,
            denominator: denominator*other.denominator);
    }
}
```

(*) 6. Extend your `Complex struct` to be `ArgandDiagramable` and test your code. (Note this can only be done in Playgrounds.)

(*) 7. Do you think it is better to add to `Complex` by placing functions in the structure itself or into extensions? Explain why.

(*) 8. Extend your `Complex struct` to be able to answer every question in your maths problem sheet (check your code produces the correct answers). Once you have done this seek additional problems that may be used to help you improve the functionality of the `struct`.

(**) 9. So we have used real and imaginary to store the state of a complex number. We could have used the modulus and argument instead from

$$z = a + \mathrm{i}b = |z| \exp(\mathrm{i}\theta) = |z| \left(\cos\theta + \mathrm{i}\sin\theta\right)$$

meaning $a = |z|\cos\theta$ and $b = |z|\sin\theta$. Under which circumstances is modulus argument form better than real and imaginary parts for storage and why? Check your hypothesis by writing and running suitable test code (this means you need to write another `Complex struct` using modulus argument form but with the same capabilities as your existing code).

Reading list

1. Mathematical Methods in the Physical Sciences, Mary Boas, Chapter 2.

References and database links

`https://docs.swift.org/swift-book/LanguageGuide/TheBasics.html`
`https://docs.swift.org/swift-book/LanguageGuide/ControlFlow.html`