

ASGS1115 - Support for Science

Semester 1 - 2019

LAB week 6: Higher Order Function

Main ideas of Higher-Order Functions:

- Functions can be returned by another function
- A function can be an input to other function

Outputting Function

Load Haskell script FuncOut.hs which contains the following Haskell function:

```
1  -- polynomial degree 2, input 0-th, 1-st, and 2-nd coefficients
2  createPolyTwo :: (Num x) => x -> x -> x -> (x -> x)
3  createPolyTwo a0 a1 a2 = polytwo
4      where polytwo x = a0 + (a1*x) + (a2*(x^2))
```

Try to figure out what the following expression based on `createPolyTwo` function does before try running them on GHCi shell.

Try to figure out what mathematical functions can be represented by it.

```
> a = createPolyTwo 1 2 3
> (createPolyTwo 3 2 1) 0
```

Challenges:

- try to define a Haskell function `funcType` that returns a function based on the different string input:
 - `"identity"` as $f(x) = x$
 - `"inverse"` as $f(x) = \frac{1}{x}$
 - `"logistic"` as $f(x) = \frac{1}{1+e^{-x}}$

Inputting Function

Load Haskell script FuncIn.hs, which contains the functions below:

```

1  -- input a function of area of shape and its single argument
2  area :: (x -> x) -> x -> x
3  area f a = f a
4
5  -- area of square
6  square :: (Num x) => x -> x
7  square x = x*x

```

Try figuring out the outputs of the following expression before running them on GHCi:

```

> (area square) 2
> sqar = area square

```

Challenges:

- try to define a Haskell function to be inputted to `area` for the following shapes:
 - circle
 - right-angled triangle (equal sides)
- define the following functions as well, and modify `area` as necessary:
 - rectangle
 - right-angled triangle (non-equal side)
 - trapezoids