Solution  Functional Programming

- Exercises are given every week on the PL page of the SCG website
  (http://scg.unibe.ch/teaching/pl)
- Solutions to each assignment must be sent to mohammadreza.hazhirpasand@inf.unibe.ch
- The solutions of the assignments are to be delivered before every Thursday at 5 PM. Solutions
  handed in later than the specified time will not be accepted. In case of serious reasons send an
  e-mail to mohammadreza.hazhirpasand@inf.unibe.ch

Exercise (6 points)

- Explain why the following piece of code does not raise an error. (1 pts)
  \[
  \text{func1 5 z = 33} \\
  \text{func1 y z = y} \\
  \text{func1 50 (sqrt(-5))} \\
  \text{- output is 50}
  \]

  \textbf{Answer:} \\
  This is allowed because of lazy evaluation. The incorrect argument here is not going to be evaluated.

- Define the following small program in three different functions with pattern matching, guards, and
  lambda expression. (1.5 pts)
  \[
  \text{if } n = 0 \text{ then return -1} \\
  \text{else return } n \times 2
  \]

  \textbf{Answer:} \\
  \[
  (\text{s\ } \rightarrow \text{if } s == 0 \text{ then -1 else } s \times 2) \ 0 \\
  \]
  \[
  \text{gpdef } n \mid n == 0 = -1 \\
  \mid n /= 0 = n \times 2
  \]
  \[
  \text{pmdef } 0 = -1 \\
  \text{pmdef } n = n \times 2
  \]

- Define a function that accepts a list as an argument and returns the sum of all the members of the
  given list. (1.5 pts)

  \textbf{Answer:} mh8 (x) = if x == [] then 0 else head x + mh8(tail(x))
Define a function `firstNCatalan n` in Haskell that calculates and returns the result as a list containing the first `n` Catalan numbers. Catalan numbers are calculated based on the formula

\[ C_n = \frac{(2n)!}{(n+1)n!}, \quad n \geq 0. \] (2 pts)

**Answer:**

```haskell
fac n
| n == 0 = 1
| otherwise = n * fac (n-1)
catalan n
| n >= 0 = fac (2*n) / (fac n * fac (n+1))
firstNCatalan n = [catalan x | x <- [0..n]]
```