Prof. O. Nierstrasz
Mohammadreza Hazhirpasand, Joel Niklaus

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)

  func1 5 z = 33
  func1 y z = y
  func1 50 (sqrt(-5))
  – output is 50

  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)

  if n = 0 then
  return -1
  else
  *return n * 2*

  Answer:

```
(s\ ->if s == 0 then -1 else s * 2) 0
----
gpdef n | n == 0 = -1
| n /= 0 = n * 2
----
pmdef 0 = -1
pmdef n = n * 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)

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

  Answer:

```
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]]
```