Assignment #3

Name: ______________________ ID: ______________________

This assignment has **3** questions, for a total of **25** marks.

Question 1: **Named functions** .... 10 marks

Consider extending STLC with named function, call this language STLCN. A program is no longer a term, but a collection of named functions ($P ::= \varnothing \mid P; F$). A named function defines a function name, a parameter of a certain type and the function body ($F ::= f(x) \mapsto t$). A function body is a term. Terms now must include new constructs to call other functions $t ::= \cdots \mid \mathit{call}\ f$.

Define the COS judgements for STLCN. Define the primitive reduction rules for STLCN as well as evaluation contexts. Define the typing judgements for STLCN, starting from how to determine when a program is well typed. Define the typing rules for STLCN.

Some primitive reductions and typing rules from STLCN will be similar to those of STLC. To avoid duplicates, name all the STLC rules that have analogous ones and show how to convert two of them to the new forms only.

STLCN should still be safe, i.e., it should not get stuck trying to call a function with a parameter of the wrong type, or calling a function that does not exist. However, STLCN is not normalising: there may be functions that mutually call each other and thus diverge.

Question 2: **Private and public functions** .... 10 marks

Extend STLC so that each program not only defines a list of named functions, but also a list of imports and exports. Imports are signatures (names and types) of functions that the program does not define. Exports are names of functions that the program defines. Change the typing rule for programs to reflect this.

Define the linking two programs $(P_1 + P_2)$, which yields the program obtained by joining the two. Linking is only possible when both programs are well-typed and when they fulfill each other's exports with their own imports.

Question 3: **Progress cases for pairs** .... 5 marks

Write the proof for the progress theorem for the following cases related to pairs: $t \equiv \langle t_1, t_2 \rangle$ [3], $t \equiv t_1.1$ [2].