Question 1 - Application of reduction and typing rules for MINIJAVA  Let $P$ be the following MINIJAVA program

```java
class A {
    int f;
    int m (B b, int x) {b.f=x;}
}
class B {
    int f;
    int m (A a,int x) { a.m(this,x);}
}
```

1. Show the class type environment $\Gamma_P$ which can be extracted from $P$.
2. Prove, by applying the typing rules, that the following expression $e$

   $\text{new } B.m(\text{new } A,7)$

   is well-formed in the class type environment $\Gamma_P$ and the empty local environment.
3. Evaluate $e$ starting from the empty store.

Question 2 - Definition of new reduction and typing rules  Assume we extend MINIJAVA with the identity operator as in Java, with the following syntax

```
e ::= e_1 == e_2
```

and the standard meaning.
Add reduction and typing rules for the operator.

Question 3 - Implementation of union types  Consider the following Java implementation of functions from integers to integers. The method `apply` returns the result of applying a function to a given argument.

```java
abstract class Function {
    abstract int apply (int arg);
}
```

Define three heir classes of `Function` (with suitable constructors) which model, respectively,

1. constant functions (that is, functions which return on each argument a given fixed result);
2. functions which return, on each argument in a given interval $[n, m]$, the successor, and return 0 on other arguments.
3. functions which return the composition of two given functions $f, g$ (that is, on each argument $x$, they return $f(g(x))$).

Then, add in `Function` a method which tests whether two functions are equal in a given interval $[n, m]$. 
4 - Additional questions  Graduate students should solve one of the following additional questions, on choice:

1. Prove, by applying the typing rules, that the program $P$ in Question 1 is well-formed. Suggestion: denote by $\text{class}_A$, $\text{class}_B$ the two class declarations in $P$, and by $\text{meth}_A$, $\text{meth}_B$ the two method declarations in $A$ and $B$, respectively.

2. Assume we extend MINIJAVA with static methods as in Java, with the standard meaning. That is, we modify the syntax as follows.

$$
\text{meth} ::= \text{static}\{ Tm(T_1, \ldots, T_n) \{ e; \} \}
$$

$$
e ::= \ldots | C.m(e_1, \ldots, e_n)
$$

Show which extensions are needed in reduction and typing rules. More precisely:

(a) add reduction rules for static method call;
(b) describe how the definition of method types should be changed;
(c) add a typing rule for static method declaration;
(d) add a typing rule for static method call.