Problem 1

1.
Function subtypes of A.m:
    "X m(X y, String s);" - because X is a subtype of Object --> Java Override
    "Z m(Y y, String s);" - Java Overload?? - Z subtype of Y subtype of X subtype of Object

Doesn't work
"Y m(Object y, Object s);" - because Object is supertype of X, Y is subtype of X
which is subtype of Object -- Overload??

2.
Constructor Overriding is never possible.
This is because even thought the constructor looks like a method, the name should be
a class name an have no return value. So there would be no practical way to override
it.

3.
From the lecture notes, we know that no new exceptions should be thrown, unless the
exceptions are subtypes of exceptions thrown by the parent.

(3) FileNotFoundException --> Subtype of IOException

(2)RuntimeException - May also work because it is an unchecked exception, so if B's
specification is weaker than A, then it can
be thrown.

4.
We say that (class) B is a true subtype of A if B is a subclass of A and has a stronger
specification than A.
A specification S2 is stronger than or equal to a specification S1 if:
    -- S2's precondition is weaker than or equal to S1's,
    -- S2's postcondition is stronger than or equal to S1's, for the states that satisfy
    S1's precondition.

Triangle - Isoc. Triangle
Isoc Subtype of Triangle -- stronger post condition so Isoc. has stronger specification
-- TRUE SUBTYPE

Vertebrate --> Squid
Subtype and Stronger post condition, TRUE SUBTYPE
Vertebrate --> Human
Subtype and stronger post condition, TRUE SUBTYPE

Bicycle --> MountainBike
TRUE Subtype because it is just adding another variable, so it will have the same preconditional value range, and the same postconditional range.

Account --> Concurrent account
Not a true subtype because it throws a more specific exception than specified in the parent. (from lecture notes)
No new exceptions should be thrown, unless the exceptions are subtypes of exceptions thrown by the parent. (from lecture notes)