15 & 16 March 2023
Problem Set #7
Variable Capture

1. Complete the method `and` that takes in two `Predicate` objects `p1` and `p2` and returns a new `Predicate` object that evaluates to `true` if and only if both `p1` and `p2` evaluate to `true`.

   ```
   <T> Predicate<T> and(Predicate<T> p1, Predicate<T> p2) { ... }
   ```

2. Study the following program fragment.

   ```
    1 abstract class A {
    2     abstract void g();
    3 }
    4
    5 class B {
    6     int x = 1;
    7
    8     void f() {
    9         int y = 2;
   10
   11         A a = new A() {
   12             void g() {
   13                 x = y;
   14             }
   15         };
   16
   17         a.g();
   18     }
   19 }
   ```

   Now suppose the following is invoked:

   ```
   B b = new B();
   b.f();
   ```

   Sketch the content of the stack, heap and metaspace *just before* the statement in line 17 is executed. Label the values and variables/fields clearly. You can assume `b` is already on the heap and you can ignore all other content of the stack and the heap before `b.f()` is called.

3. You are given two functions $f(x) = 2 \times x$ and $g(x) = 2 + x$.

    (a) By creating an abstract class `Func` with a public abstract method `apply`, evaluate $f(10)$ and $g(10)$.

    (b) The composition of two functions is given by $f \circ g(x) = f(g(x))$. As an example, $f \circ g(10) = f(2 + 10) = (2 + 10) * 2 = 24$. Extend the abstract class in question 3a so as to support composition, i.e. `f.compose(g).apply(10)` will give 24.

    (c) Now re-implement the `Func` abstract class as generic abstract class `Func<T,R>` with the corresponding re-definitions of `apply` and `compose` methods.