CSC 203 Quiz 5 **Your name**: ____________________

May 21, 2025

Note: You may use your notes or my notes from my website. No other resources may be used during this quiz.

Question 1. (5 points) Suppose you have a class `Cat` with three instance variables:

- `String name`
- `int age`
- `boolean isUpForAdoption`

(You may assume these variables have public getter methods.)

We would like to define a `Comparator` that would order two cats in **descending** order by `age`, and break ties in **ascending** order by `name`.

Fill in the blanks to implement the `Comparator`.

`Comparator<Cat>` comp =

`Comparator.comparing(`*`Cat::getAge`*`)`

.*`reversed()`*

.*`thenComparing`*(*`Cat::getName`*);

Question 2. (4 points) You are looking for an older cat to adopt. Fill in the blanks below to create a lambda expression to help decide if a `Cat` is over the age of 4 and is up for adoption. (For example, we might use this lambda in a filter function.)

Complete the function below.

`Predicate<Cat>` olderAndAdoptable =

`c` -> *`c.isAdoptable() && c.getAge() > 4`*;

Question 3. (3 points) Fill in the blanks below to define a generic `compose` method that takes in an input of type `I`, applies the first function to it (`func1`), and then applies the second function (`func2`) to the result.

For example, `compose("CSC", s -> s.length(), n -> n % 2 == 0)` should evaluate to `false`.

```
public static <I, O, U> U compose(I input, Function<I, O> func1, Function<O, U> func2) {
    return func2.apply(func1.apply(input));
}
```