

NAME: _____

CS 330 Final
Spring 2015

Doodle here.

On the following pages, write things that are true and relevant.

1. *Outputrrss*

Bjarne Stroustrup introduced a new I/O system for C++ because C's wasn't type-safe. In C, a statement like `printf("%d", i)` was not part of the language proper, and the compiler could not assert that the `%d` matched the type of `i`. In C++, we get type-safe I/O by overloading the `operator<<` method:

```
ostream &operator<<(ostream &out, int i);
ostream &operator<<(ostream &out, const Node &i);
ostream &operator<<(ostream &out, const RTree &i);
ostream &operator<<(ostream &out, const PowerUp &i);
ostream &operator<<(ostream &out, const Sandwich &i);
```

What is Java's strategy for typesafe output of custom objects in the `PrintStream` class (of which `System.out` is an instance)? What language feature(s) does the Java strategy exploit?

2. *RY*

Write function `repeat` in Scala that executes a block some specified number of times:

Code	Output
<pre>repeat (2) { println("Grumble") }</pre>	Grumble Grumble

The number of iterations is determined exactly once before the loop starts. Modifications to any variables of the condition expression inside the block do not alter the number of iterations:

Code	Output
<pre>var n = 4 repeat (n) { print(".") n *= 2 }</pre>

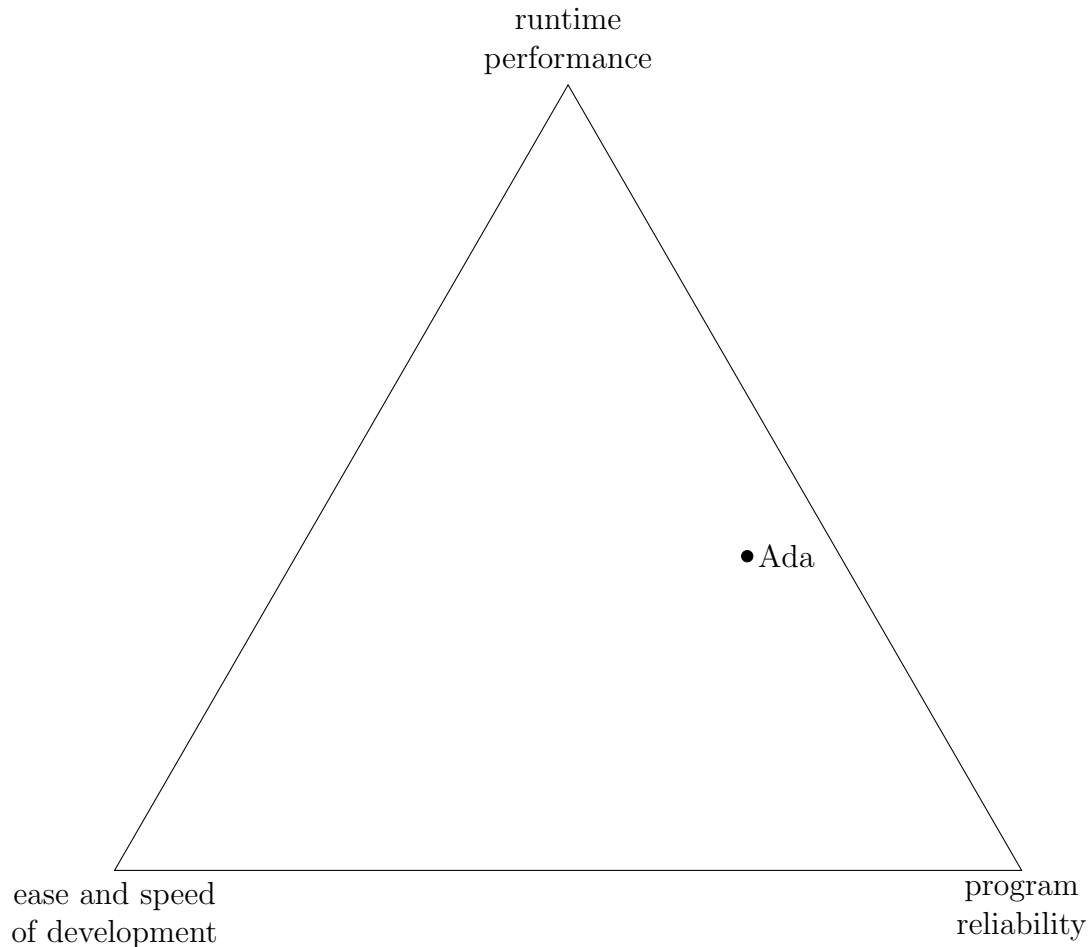
What Scala features does your solution exploit?

4. *Fallback*

Java 7 and earlier versions do not support higher-order functions. How then does one register a callback in these versions of the language?

5. *Triangle of Concerns*

Consider the following triangle, which represents the tension developers must navigate when choosing a programming language. Do we go for speedy execution? Quick development? Or a program that is provably reliable? Typically, a gain in one of these qualities means a loss in the others. Plot each of the 8 languages listed on the next page where you believe it belongs on this triangle of concerns. A language that focuses on program reliability, like Ada, will appear near the reliability vertex, but it can also produce fairly fast machine code (with some runtime checks to maintain reliability). On the next page, briefly justify your placements. A language's relative location is more important than its absolute location.



Java

C++

C

Ruby

JavaScript

Haskell

Scala

Shell

6. Consider the type signatures of three canonical higher-order functions in Haskell:

```
map :: (a -> b) -> [a] -> [b]
filter :: (a -> Bool) -> [a] -> [a]
foldr :: (a -> b -> b) -> b -> [a] -> b
```

Use these functions to implement the following functions, whose affordances may be inferred from the example invocations:

(a) `retreat [5, 10, 20] → [4, 9, 19]`
`retreat [0, -10] → [-1, -11]`

(b) `plusside [-5, 0, 5, 10] → [5, 10]`
`plusside [100, -100] → [100]`

(c) `contains "abcdefg" 'z' → False`
`contains [0,2..10] 6 → True`