Exam 1 Review

Scala

Georgia Tech

General Scala

- How do you pronouce Scala?
- The main version of Scala compiles to bytecode for which platform?
- How can you run Scala code?
- What is Scala's primary build tool?
- What is the highest supertype of all Scala objects?
- What is the lowest subtype of all Scala objects?
- What is the difference between == and `eq`?
- Which packages are implicitly imported in every Scala source file?

Georgia Tech

Values, Variables and Control Structures

- What is the value and type of `s` below?

```
1  val moritz = 1865
2  val s = {
3    "Max"
4    moritz
5  }
```

- What is the value and type of `result` below?

```
1  val result = if (true) "blue" else 2
```

- What is the difference between a `val` and a `var`?

Georgia Tech

Collections

- Write an expression that gives the number of elements in `xs`.

```
1  val xs = Set( ... )
```

- Write an expression that creates a `Map` referenced by a `val` named `langs` such that

```
1  scala> langs("Lisp")
2  res0: String = John McCarthy
3
4  scala> langs("Java")
5  res1: String = James Gosling
6
7  scala> langs("Pascal")
8  res2: String = Niklaus Wirth
9
10 scala> langs("Scala")
11 res3: String = Martin Odersky
```

- Write a for-comprehension that uses `langs` from the previous question to create a `Seq` of the last names of the values in `langs` that is, `List(McCarthy, Gosling, Wirth, Odersky)`.

Functions

- Write a function called `mean` that takes a variable number of `Double` parameters and returns their arithmetic mean (sum divided by number of numbers).
- Given `xs: List[Int]`, write an invocation of the `filter` method on `xs` that passes a function literal which selects only the odd numbers in `xs`.
- Write a function named `listToString[T]` that takes a single `List[T]` and returns a string representation of the `List[T]`. The `listToString` function should have a single expression which calls a nested helper function which is recursive and uses pattern matching to recursively accumulate a string representation. Your helper function may use an if statement instead of pattern matching for partial credit.

Georgia Tech

Classes and Objects

- Write the minimal Scala definition of a (non-case) class named `Item` which has two fields, `name` of type `String` and `hauptstadt` of type `String`.
- Write an `equals` method for the `Item` class above using the recipe we discussed in class.
- Write a `hashCode` method for the `Item` class above using the recipe we discussed in class.
- Write a companion object for the `Item` class above with a factory method that allows us to create an `Item` object with expression like `Item("Key Lime", 3.14)` (leaving off operator `new`).

Georgia Tech

Inheritance

Given the `Person` class below, write the minimal non-final subclass of `Person` named `Employee` that adds a mutable `salary: Double` field and initializes the fields defined in `Person`.

```
1  class Person(val name: String)
```

Georgia Tech

Case Classes and Pattern Matching

- Given the classes below, write a function named `next` which takes a single parameter `fußgängerampel: AmpelMann` and returns the next `AmpelMann` in the `Grün -> Rot -> Grün` cycle. Your `next` function body should be a single `match` expression.

```
1 sealed trait AmpelMann
2 case object Grün extends AmpelMann
3 case object Rot extends AmpelMann
```

Georgia Tech

Algebraic Data Types

- Write minimal traits/classes/case classes such that the following function would compile without warning or error, but if you removed one of the cases it would compile with a "match may not be exhaustive" warning.

```
1  def hauptstadt(land: Bundesland) = land match {
2    case land: Berlin => "Berlin"
3    case land: Brandenburg => "Potsdam"
4    case land: MecklenburgVolpommern => "Schwerin"
5    case land: SachsenAnhalt => "Magdeburg"
6    case land: Sachsen => "Dresden"
7    case land: SchleswigHolstein => "Kiel"
8    case land: FreieHansestadtHamburg => "Hamburg"
9    case land: HansestadtBremen => "Bremen"
10   case land: Niedersachsen => "Hannover"
11   case land: NordrheinWestfalen => "Düsseldorf"
12   case land: FreistaatThüringen => "Erfurt"
13   case land: Hessen => "Wiesbaden"
14   case land: RheinlandPfalz => "Mainz"
15   case land: Saarland => "Saarbrücken"
16   case land: BadenWürttemberg => "Stuttgart"
17   case land: FreistaatBayern => "München"
18 }
```

Georgia Tech