Scala: A **Scalable language**

- Multi-paradigm language
- Small core
- Designed with scalability in mind
- Runs on the Java virtual machine
- Interoperates seamlessly with Java
Scala: A Scalable language

• Builds new constructs from basic, simple components
• Can reuse and adapt components
• Can add libraries that appear as language extensions
  • delayed argument evaluation
  • infix syntax for methods
Scala: A **Scalable language**

- Supports **programming in the small**
  - interpreter with REPL
  - scripting abilities
  - concise syntax
- Supports **programming in the large**
  - Classes, packages, libraries, ...
  - Static-typing
  - separate compilation
Multi-paradigm language

• Integrates features of
  • object-oriented
  • functional
  • concurrent
languages

• The three programming styles complement one another
Purely Object-Oriented

- Every value is an object
- Types and object behavior are defined by classes and traits
- Classes are extended by subclassing and mixin-style composition
- Operators are methods
  - 3+2 is syntactic sugar for 3.+(4)
Highly Functional

- Every function is a value, and so an object
- Almost everything is an expression
- Anonymous and higher-order functions
- Curried functions/partial application
- Lazy evaluation
- Pattern matching
Concurrent

• Supports the **Actor model**
• Simple but expressive and scalable
• Based on message passing between asynchronous actors
• Appears like a native aspect of the language
• In reality, just a library built on top of JVM threads
Expressive and Concise

• very powerful constructs
• statically typed but rarely requiring type annotations
• very little boilerplate code
• higher-level than mainstream OO languages
• intuitive and readable syntax
Java

```java
class MyClass {
    private int index;
    private String name;
    public MyClass(int index, String name) {
        this.index = index;
        this.name = name;
    }
}
```

Scala

```scala
class MyClass(index: Int, name: String)
```
Expressive

Java

```java
boolean nameHasUpperCase = false;
for (int i = 0; i < name.length(); ++i) {
    if (Character.isUpperCase(name.charAt(i))) {
        nameHasUpperCase = true; break;
    }
}
```

Scala

```scala
val nameHasUpperCase = name.exists(_.isUpperCase)
```
Resources

A comprehensive starting point is Scala's official website:

http://www.scala-lang.org

See also the Resources section on the course website