Define local variable.

A variable defined inside a block. This does not include parameters, which are similar.

What's a synthetic class?

A class generated automatically by the compiler rather than being written by hand by the programmer.

What are variables?

What can they be?

Named entities that refer to an object.

They can be either var or val.

Give two definitions of value.

- The result of computation or expression.
- Formally, the image of an object in memory.

What are helper functions?

How are they usually implemented in Scala?

Functions that service nearby functions.

They are usually implemented as local functions in Scala.

Define referential transparency.

The property of functions that are independent of temporal context and have no side effects.

Referentially transparent functions' results can be replaced with an invocation of the function without changing the program's results.

What is the functional style?

What are its characteristics?

Style emphasizes functions and evaluation results, de-emphasizing the order in which operations occur.

Characterized by:
- Passing function values into looping methods.
- Immutable data.
- Methods with no side effects.

Discuss reference and referring.

Java abstraction of a pointer, which uniquely identifies an object on the JVM's heap.

Variables hold references, but "referring" is more abstract than "holding a reference" because a variable of type `AnyVal` may hold a reference to an Integer, but may not if a primitive int is being used.

Define the uniform access principle.

Variables and zero parameter functions should be accessed using the same syntax.

Define by-name parameter.

What are all others?

A parameter marked with => before type.

Corresponding arg is not evaluated before the method is invoked, but each time the parameter is referenced by name inside the method.

All others are by-value.

Define function literal.

What does the spec call them?

A function with no name, specified with function literal syntax.

The spec calls them "anonymous functions".

Define statement.

An expression, definition, or import.

i.e., things that can go into a template or a block in Scala source code.

Define block.

What doesn't qualify as a block?

Encapsulation construct for which you can only see side effects and a result value?

Curly braces of class don't qualify because members are visible.

Define local functions.

Functions defined inside a block.

Define method.

Functions (in the general sense) that are members of a class, trait, or singleton object.

Define expression.

Any bit of Scala that evaluates to (results in) a value.

Define predicate.

A function with Boolean results.

Define static type.

It restricts the possible values to which a variable can refer, or an expression can produce, at runtime.

Describe define.

Give a name and implementation, hence abstract members are declared and not defined.

Define auxiliary constructor.

Extra constructors defined inside curly braces of class definition, which look like methods named "this" with no result type.

When can objects be reclaimed?

When they are unreachable.

This is not the same as unreferenced.

Define bound variable.

A bound variable of an expression is a variable that's both used and defined inside the expression.

Define free variable.

A free variable of an expression is a variable that's used but not defined in an expression.

```
(x:Int) => (x,y)
```

y is free.

Define procedure.

A function with result type of Unit. It's executed only for its side effects.

Define template.

The body of a class, trait, or singleton definition.

It defines type signatures, behavior, and initial state.

Define member.

What is this in contrast to?

Any named element of the template of a class, trait, or singleton object.

It can be accessed with the name of owner, a dot, and its simple name.

- Also, a class/singleton/trait is a member of the package in which it was defined.
- By contrast, a local variable/function is not a member of its surrounding block.

Define target typing.

A form of type inference that takes into account the type that's expected.