Nick Stone

**Scala Questions Answered.**

**Question 1. Is it a strong type or weak type language?**

Since Scala is backwards compatible with Java, Scala shares Strong type checking. This can be difficult to understand since even though Scala Is an interpreted language the compiler will check and make sure the variables are of the same type before operations will be processed. If two incompatible types are checked an error will result.

**Question 2. What is Scala’s main purpose?**

The purpose of Scala is to build an application of any type and have it been able to scale with your user base. While Scala was originally created for web applications today Scala is used to process big data and for data visualization.

**Question 3. Are there readability issues due to backwards compatibility in Scala?**

When looking at a Scala project that invokes the ability to use Java code and libraries the code does become much more difficult to read. This is due to the fact that when importing a library in Scala the syntax of the code is significantly different which makes the code overall harder to read but is a clear indication to the user that a library has been called.

**Question 4. Are there often bugs that result from automatic return?**

The use of the automatic return can lead to bugs, but since the use of the automatic return function will be usually be invoked by functional programming styles this should not produce more bugs since functional programming usually leads to less bugs in the code. As well as the programmer still has the ability to use the return key word, this meaning the problem can be eliminated entirely if the programmer feels that the automatic return takes away from program readability.

**Question 5. What features of Scala help make it more easily scalable than Java or C?**

Since Scala has the ability to write functional programs a programmer can use functional programming tactics when dealing with areas of a program that will be prone to bugs. What this means is that when dealing with something such as big data they can easily manipulate data much more efficiently. Among fewer bugs Functional programming allows a programmer to utilize quick data manipulation tactics.

**Question 6. If you’re not able to change object declared by “Val” Why did Scala think “Val” was useful?**

The use of the Val function works the same as the use of the Final key word in Java. All this keyword does is not allow the programmer to change the value of an object or primitive type. In some scenarios that will be programs that variables and objects stay the same throughout the duration, suppose we are manipulating big data and we want a given variable to never change but we constantly use this variable we do not want the variable integrity to mistake on the programmer’s part.

**Question 7. Do you think the backwards compatibility with Java limits Scala at all instead of being its own language?**

The use of backwards compatibility with Java only improves Scala’s ability to be seen as its “own language”. With Scala functional programming and use of Java libraries Scala can be a tool that can be used for anything Java can be used for as well as applications such as big data manipulation that will rely on functional programming practices. Furthermore, with the use of both functional and object-oriented programming styles it allows programmers from different backgrounds to work together and build applications under one language. This allows for software applications to be written in one language, instead multiple languages used throughout the application.