In 1998 at the ACM OOPSLA conference Guy Steele gave an influential speech about coding languages, and their characteristics. During the speech, Steele used an interesting technique where he used his spoken vocabulary as if it were a programming language. He did this by starting with a base set of words that he defined, and then later introducing more words into his vocabulary and defining those using previously defined terms. This technique served Steele in his talk by how it gave a very nice example of a programming language and its progression through time. He really backed it up when he stated that a programming language should be neither big nor small, but start out small with the ability to grow and adapt to suit its user’s needs. Steele explained that in order for a programming language to come into full fruition, you could neither treat it as a cathedral or a bazaar. A cathedral being too bound by the rules laid out by those who own it, and a bazaar being too open for change by those who use it. Instead it needed to possess the same properties as that of a shopping mall. AKA something that does not have all the entire selection in the world, but instead has particular selections that it does very well. Steele emphasized this by giving examples of commonly used data types such as vectors, and asking the audience if they should be added to Java. He gave the answer when he stated that each individual one could be added, but none of them should. These individual data types are very user exclusive, and catered to too few a user base to be added, and if they all were in order to accommodate everyone, they would be too much for the users of java, and as a language it would be harder to learn. The self-acclaimed high point of Steele’s talk was when he gave a quote by Christopher Alexander. This quote explained that master plans as far as languages go are not good. They alienate the users from the designers and create a language that was not tailored to the community as well as it could have been. In this way programming languages must be made as a collaborative effort by the community, and not for individual’s purposes. Although this speech was given more than 15 years ago, I believe it is almost better to watch it now, because it gives its viewers a chance to reflect over what they have learned about programming, and what they see in languages today, and witness first-hand how well this speech predicted the future of programming. This talk really brought to light the vastness of the programming world, and how every single one of us affects it. Through our collaborative use of any one of the programming languages, we affect the future of each and every language we use. I would highly recommend that future classes be asked to watch this video, as I see it as an important eye-opener to the intricacies of the world of computer programming languages.