1. High languages provide a nice abstraction from assembly language. One benefit to high languages is to abstract the developer from knowledge of registers and other things such as the stack and frame pointer. High languages also are more human readable and thus can be easier to program in. It can make sense to program in assembly if a developer is interacting directly with the architecture of the computer.
2. Declarative programming is written to tell the program what to accomplish but not how to accomplish it. This is common in functional languages and it reduces side effects. Imperative programming is written in a way to tell the compiler explicit instructions.
3. A) The first programming language I learned was BASIC. It was a course taught in high school. I think the main reason it was taught was because the ease of use in terms of UI. We were able to drag and drop items to create UIs, a lot simpler than Java Swing.  
     
   B) I wish Java was dynamically typed. I also wish the convention wasn’t to make Java so verbose. Chaining calls in Java can get tedious and annoying.
4. Cost could be affected by the goal of the programming language. If the purpose of the language is to simplify quantum mechanics, it could take longer to write and therefore become costlier. Another aspect of cost could be the choice between dynamically and statically typed languages. If a language is dynamically typed it may take more work in the compiler to ensure a program is actually legitimate.