HW1 Solution

1) Some arguments against having a single language for all programming domains are:

- The language would necessarily be huge and complex;
- Compilers would be expensive and costly to maintain;
- The language would probably not be very good for any programming domain, either in compiler efficiency or in the efficiency of the code it generated.
- More importantly, it would not be easy to use, because regardless of the application area, the language would include many unnecessary and confusing features and constructs (those meant for other application areas). Different users would learn different subsets, making maintenance difficult.

2) One of the main arguments is that regardless of the cost of hardware, it is not free. Why write a program that executes slower than is necessary. Furthermore, the difference between a well-written efficient program and one that is poorly written can be a factor of two or three. In many other fields of endeavor, the difference between a good job and a poor job may be 10 or 20 percent. In programming, the difference is much greater.

3) The main costs associated with software development are:

- Cost of training programmers
- Cost of writing programs
- Cost of compiling programs
- Cost of executing programs
- Cost of maintaining programs
- Cost of porting programs

4) The argument for typeless languages is their great flexibility for the programmer. Literally any storage location can be used to store any type value. This is useful for very low-level languages used for systems programming. The drawback is that type checking is impossible, so that it is entirely the programmer's responsibility to insure that expressions and assignments are correct.

5) The main motivation for the development of PL/I was to provide a single tool for computer centers that must support both scientific and commercial applications. IBM believed that the needs of the two classes of applications were merging, at least to some degree. They felt that the simplest solution for a provider of systems, both hardware and software, was to furnish a single hardware system running a single programming language that served both scientific and commercial applications.