CSCI 485

Assignment 1

1. Why were the first programming languages developed? What problem did they intend to solve?

It allowed humans to interact with machines and have them perform operations. The problem they wanted to solve was the time it took for a programmers code to be interpreted/executed.

1. What’s the difference between an assembler and a compiler?

Compiler had a one-to-one from assembly to machine code. An assembler had a more complicated process which was machine independent code to machine/assembly.

1. What was the first programming language? What were the initial perceptions, and how did they change?

The first programming language was Fortran. It was seen as slower than writing out the original language which made it less efficient. Fortran changed so that it was able to optimize the compiler to make the og language almost the same efficiency.

1. Why are there so many programming languages? What are the factors of successful programming languages?

Each language has a unique reason for being developed. These factors can be personal preferences, power and cost, different syntax usage, and specific application domains.

1. What are declarative languages? What are the categories of them? Name two of each category.

Declarative languages are languages that have an objective rather than being told how to run. The three categories are functional, dataflow, and logic. Two from each are ML and Haskell, Val and Id, and SQL and spreadsheets.

1. What are imperative languages? What are the categories of them? Name two of each category.

Imperative languages are languages that are defined in a step-by-step process. The categories are Von Neumann, Object-oriented, and Scripting. Two from each are C and Ada, Eiffel and Java, and Python and PHP.

1. What’s the difference between an interpreted and a compiled language? How well do they perform? How do you execute the target application?

Interpreted language act as virtual machines with machine code while compilers make executable machine code from source code. Compiled code has a better performance than interpreted. Compilers take the source code, turns it into machine code, then outputs it. Interpreters take the source code, becomes a “virtual machines” where they make the source code to machine code.

1. Name 5 languages built on C.

C++, c#, Java, Objective C, and Concurrent C.

1. What are linkers?

Linkers are applications that allow the user to connect their code to different libraries.

1. What are preprocessors?

Preprocessors is a tool designed to help perform transformations by using simple pattern matching.

1. What is bootstrapping? How does it work?

Bootstrapping is the technique of using an interpreter and building upon it to have more sophisticated versions later on. This works by continuously implementing it and added upon it.

1. What is just-in-time compilation?

A just-in-time compiler is a compiler that translates the java bytecode into machine language just before each execution of the program.