

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

- The first programming languages were developed to address the limitations and complexities of manually writing machine code. The development of programming languages solved the problem of new computers valuing programmer's time.

2.) What's the difference between an assembler and a compiler?

- An assembler transforms assembly to machine code: one-to-one
- A compiler transforms machine independent to assembly/machine code: many-to-one

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

- Fortran was the first programming language.
- The initial perceptions which was that Fortran was slow to take over however overtime it was realized it more efficient to use.

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

- There are many programming languages because preferences in coding style and the creation of special applications. Successful programming languages depend on the ease of modifying source codes.

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

Aare "the objective rather than how to run".

- ◆ Functional: ML and Haskell
- ◆ Dataflow: Id and Val
- ◆ Logic, constraint-based: Prolog and spreadsheets

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

Imperative Languages are "code that defines step-by-step process of execution".

- ◆ Neumann: C and Fortran
- ◆ Object-oriented: Smalltalk and Java,

◆ Scripting: Perl and Python

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

-Interpreted languages exist while code is running. Compiled languages go away after machine code is generated. Interpreted debug messages better however compiled language generally has better performance. To execute target application interpreters often use a preprocessor to convert source code into an easy-to-read format and compiled use another application called a linker to merge code from external libraries.

8.) Name 5 languages built on C.

-C++, Java, Objective C, Perl, Python

9.) What are linkers?

-Statically linkers are libraries compiled into the applications. Dynamically linkers are libraries not compiled into the application and must exist on the target machine.

10.) What are preprocessors?

-Tools or programs that process code before it is compiled or executed.

11.) What is bootstrapping? How does it work?

-The process of starting a computer or system. When starting the system its core components and the necessary software are loaded to allow the system to work.

12.) What is just-in-time compilation?

-A technique used in computer programming to improve the performance of programs.