Define *type system*.

Name a major difference between the type systems studied in the context of programming languages and those studied in the context of pure typed lambda-calculi.

What is the relationship between *terms*, *types*, and *values*.

What is problematic with the term *dynamically typed language*.

In what sense are type systems *conservative*?

What is the main research goal in the study of type systems?

The particular bad behaviors a type system is trying to prevent is called ...

A type system that successfully prevents all such behaviors is called ...

What are the benefits of type systems?

What makes a language *safe*?

Why are there virtually no unsafe dynamically checked languages?

Why can't safe languages rely solely upon the type system to guarantee safety?

Contrast *run-time type error* with the more general notion of unsafe behaviors.

What was the goal of the first type systems for programming languages?