General Concept: Types

• Terms to learn:
  – Type safe
  – Statically typed language
  – Dynamically typed language
  – Strongly typed
  – Weakly typed

Type Safety

• Type safe means that the language guarantees that a
  value of one type can’t be incorrectly used as if it were
  another type, in other words, that all expressions are
  guaranteed to be type consistent.
• Miranda, Java, Smalltalk, Scheme, Haskell, and Ada are
  examples of type safe languages.
• Fortran and C are examples of languages that aren’t
  type safe.
• Some languages for systems programming, for example
  Mesa, have a safe subset, although the language as a
  whole is not type safe.
• Weakly typed. Weakly typed means “not type safe”.

Static vs Dynamic Typing

• An orthogonal issue to type safety is when the
  types of expressions are known and the
  statements in a program are checked for type
  safety.
• We can know the type of an expression at
  compile time or not until runtime, and thus we
  can check for type safety at compile time or at
  runtime. Many languages do a mixture of these.

Static vs Dynamic Typing - definitions

• Statically typed. Statically typed means that
  the type of every expression can be determined
  at compile time. Miranda and Ada are examples
  of statically typed languages. (Scheme is not
  statically typed though.)
• Speaking more loosely, we may refer to a
  language as statically typed if the compiler can
  determine and check the types of almost all
  expressions at compile time.
• Dynamically typed. The types of expressions
  are not known until runtime. Example
  languages: Smalltalk, Scheme.

Tradeoffs

• Generally we want languages to be type safe.
• An exception is a language used for some kinds of
  systems programming, for example writing a garbage
  collector. The “safe subset” approach is one way to deal
  with this problem.
• Advantages of static typing:
  – catch errors at compile time
  – machine-checkable documentation
  – potential for improved efficiency
• Advantages of dynamic typing:
  – Flexibility
  – rapid prototyping

More Terminology about Types

• Unfortunately different authors mean different things by
  the terms “statically typed” and “strongly typed”.
• Our definition of a statically typed language implies that
  such a language is also type safe. Other authors define
  “statically typed” to just mean that the compiler can
  statically assign a type to every expression – but that
  type might be wrong.
• By this definition Fortran is statically typed.
• Strongly typed. We will equate strongly typed and
  type safe.
• For other authors, strongly typed implies type safe and
  statically typed. (Is Scheme strongly typed?)
• To avoid misunderstanding, one can describe a
  language as e.g. “type safe and statically typed”.