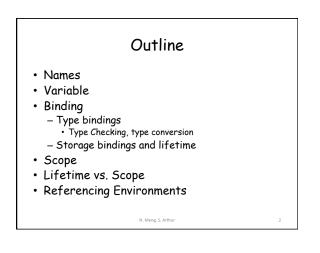
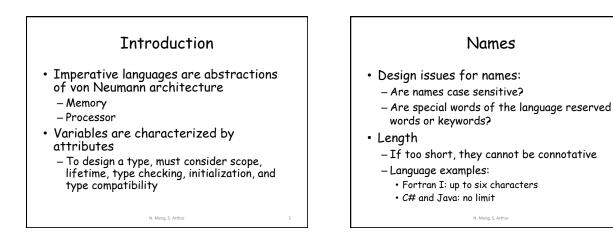


In Text: Chapter 5





Names (continued)

Case sensitivity

- Disadvantage: readability (names that look alike are different)
 - Names in the C-based languages are case sensitive

N. Meng, S. Arthur

• Names in others are not

Names (continued)

Special words

- An aid to readability; used to delimit or separate statement clauses
- A keyword is a word that is special only in certain contexts
- A reserved word is a special word that cannot be used as a user-defined name
- Potential problem with reserved words: If there are too many, many collisions occur (e.g., COBOL has 300 reserved words!)

N. Meng, S. Arthur

1

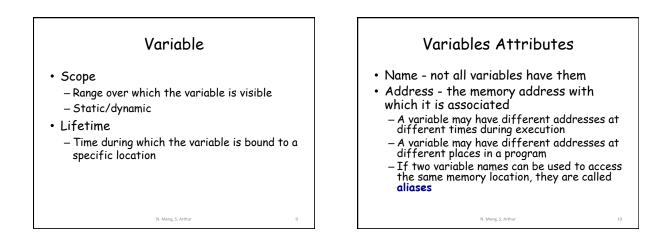
Names (continued)

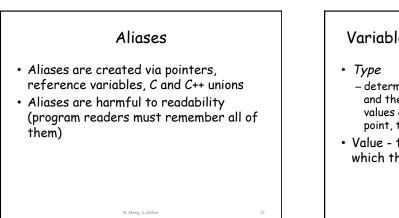
- Special characters
 - PHP: all variable names must begin with dollar signs
 - Perl: all variable names begin with special characters, which specify the variable's type
 - Ruby: variable names that begin with @ are instance variables; those that begin with @@ are class variables

N. Meng, S. Arthur

Variable

- A program variable is an abstraction of a memory cell or a collection of cells
- It has several attributes
 - Name: A mnemonic character string
 - Address
 - Points to location memory • May vary dynamically
 - Type
 - Range of values + legal operations
 - E.g., int type in Java specifies a value range of -2147483648 to 21473647, and arithmetic
 - operations for + p. Theng, 5 Atyur%





Variables Attributes (continued)

- determines the range of values of variables and the set of operations that are defined for values of that type; in the case of floating point, type also determines the precision
- Value the contents of the location with which the variable is associated

N. Meng, S. Arthur

12

14

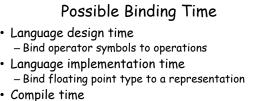
Variables Attributes (continued)

• Abstract memory cell - the physical cell or collection of cells associated with a variable

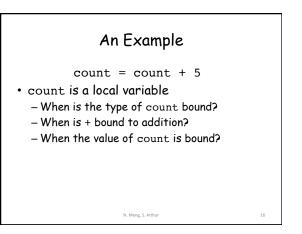
N. Meng, S. Arthur

Binding A binding is an association between two things, such as a name and the thing it names Binding time is the time at which a binding takes place

N. Meng, S. Arthur

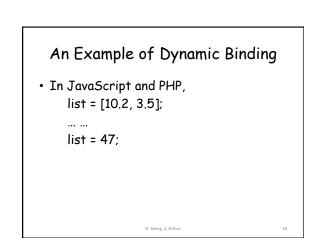


- Bind a variable to a type in C or Java
- Load time
 - Bind a variable to a memory cell (C static variable)
 Runtime
 - Bind a nonstatic local variable to a memory cell (method variables) N. Meng. S. Arthur 15



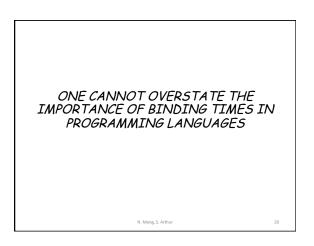


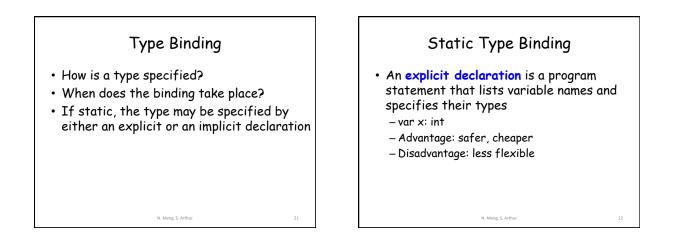
- run time and remains unchanged throughout program execution
- A binding is dynamic if it occurs during execution or can change during execution of the program



Static and Dynamic Binding • As binding time gets earlier: – execution efficiency goes up – safety goes up – flexibility goes down • Compiled languages tend to have early binding times • Interpreted languages tend to have later bindings

N. Meng, S. Arthur



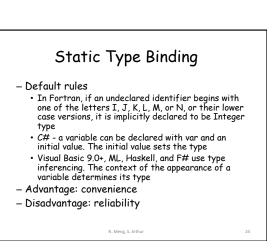


Static Type Binding

- An implicit declaration is a means of associating variables with types through default conventions, rather than declaration statements
 - First use of variable: X := 1.2;
 - X is a float and will not change afterwards

N. Meng, S. Arthur

 In C# or Swift, a var declaration of a variable must include an initial value, whose type is taken as a type of the variable

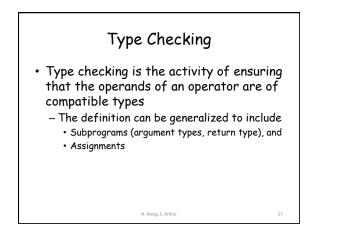


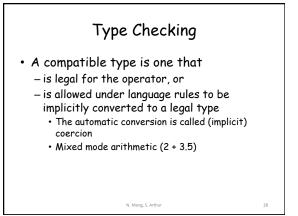
Dynamic Type Binding

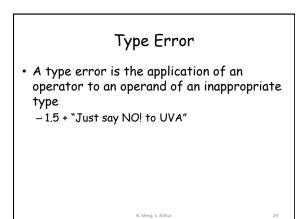
- The type of a variable is not specified by a declaration statement, nor can it be determined by the spelling of its name
 – JavaScript, Python, Ruby, PHP, and C# (limited)
- Specified through an assignment statement
 - E.g., list = [10. 2, 3.5]; (JavaScript)
 - Regardless of its previous type, list has the new type of single-dimension array of length 2

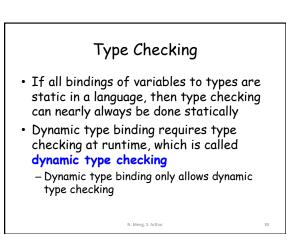
N. Meng, S. Arthur

Dynamic Type Binding (continued) Advantage flexibility (can change type dynamically) Disadvantage Type error detection by the compiler is difficult High cost Type checking must be done at runtime Every variable must have a runtime descriptor to maintain the current type The storage used for the value of a variable must be of varying size









32

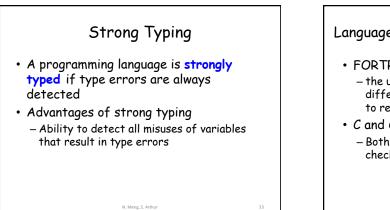
Type Checking

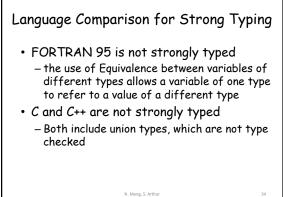
- Type checking is complicated when a language allows a memory cell to store values of different types at different times during execution
 - E.g., C and C++ unions
 - In such cases, type checking must be dynamic
- Even though all variables are statically bound to types, not all type errors can be detected by static type checking

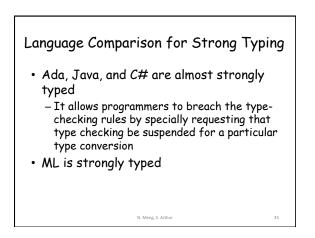
N. Meng, S. Arthur

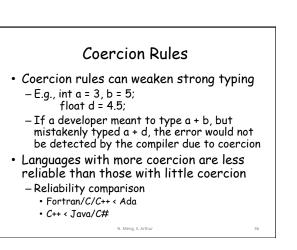
Type Checking

- It is better to detect errors at compile time than at runtime
 - The earlier correction is usually less costly
- Penalty for static checking
 - Reduced programmer flexibility
 - Fewer shortcuts and tricks are possible









38

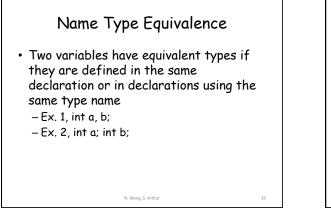
Type Compatibility

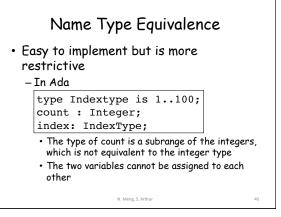
- The rules dictate the type of operands that are acceptable for each operator and thereby specify the possible type errors of the language
- Type rules are called compatibility because in some cases, the type of an operand can be implicitly converted by the compiler or runtime system to make it acceptable to the operator

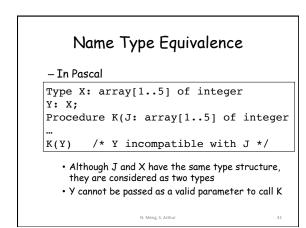
N. Meng, S. Arthur

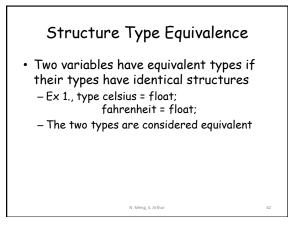
Type Equivalence

- A strict form of type compatibility compatibility without coercion
- Two approaches to defining type equivalence
 - Name type equivalence (Type equivalence by name)
 - Structure type equivalence (Type equivalence by structure)









Structure Type Equivalence

- More flexible, but harder to implement – The entire structures of two types must be compared
- Developers are not allowed to differentiate between types with the same structure

N. Meng, S. Arthur

43