Enumerations

Enumerations

- In C, enumerated types are a way to define constants.
- The syntax is similar to struct

  enum-specifier identifier { enumerator list };

  ex: enum bool_tag{ FALSE, TRUE };
- The identifiers in the list are constant ints.

Enumerations

- Values can be automatically or explicitly assigned

  enum bool_tag { FALSE, TRUE };

  FALSE is a constant int with value 0

  TRUE is a constant int with value 1
- Variables declared as type 'enum-specifier' are arithmetic integral types
- constant names across all enumerations and variable names in the same scope must be distinct. However, values need not be unique

Enumerations

- Advantages
  - automatic assignment
  - potential for type checking (check compiler ref. or C++)
  - debugger may understand the symbolic name
- Dangers
  - assumptions about type checking
  - errors in explicitly assignments
- Example

  http://faculty.washington.edu/sproedp/advc/csamples/less16.c.html