Overloading

- Using the same name and syntax to perform the same task for objects of different types.
- Example:
  ```c++
  max(5, 3) //max of two int’s
  max(3.2, 4.9) //max of two double’s
  max("abc", "abb") //max of two char*’s
  ```
- Another example?

Illegal Overloading

- Some types cannot be distinguished, so you cannot overload functions with them.
- Example:
  ```c++
  bool is_good(Beer);
  bool is_good(Beer&);
  ```
- Legal and illegal groups are detailed in Tables 6.1 and 6.2 (pp. 267-8)
Single Argument

- Compiler tries, in order, the following methods:
  - Exact match
  - Trivial conversion (Table 6.3)
  - Promotion (Table 6.4)
  - Standard conversion (Table 6.5)
  - User-defined conversion
  - Ellipsis (…) – will cover later.
  - Compilation error

Examples

User-Defined Conversions

- Conversion via constructor: if the arguments match a constructor signature, then use it.
  - Example:
    ```
    class Beer {
        Beer(int);
        bool is_good(Beer);
        bool is_good(Whiskey);
        is_good(5); // will compile
    }
    ```

Matching Multiple Arguments

- For each argument find the set of best matching functions.
- The intersection of these sets must be a single function, otherwise ambiguous call.
  - Example:
    ```
    int max(int, int); // function 1
    float max(float, int); // function 2
    float max(float, float); // function 3
    ```
    ```
    max(1.5, 1) calls function ?
    max(1, 1.5) calls function ?
    ```

Operator Overloading

- Operators (+, +=,…) can be overloaded for user-defined types.
- Can be member or non-member functions.
  - Examples:
    ```
    Beer operator+(const Beer & b, const Beer & c) {
        return Beer(b.get_name()+c.get_name(),
                    avg(b.get_taste() + c.get_taste()));
    }
    ```
    ```
    Beer & Beer::operator+=(const Beer & b) {
        *this = *this + b;
    }
    ```

Assignment Operator

- By default assignment makes a member-wise copy.
- Most classes should overload it.
  - Compare `bud` and `budLite`:
    ```
    Beer bud;
    bud = Beer("bud", 2);
    Beer budLite("budLite", 2)
    ```

Midterm Details

- On Friday, Jan 31, in class.
- Open book/notes, no computers.
- Covers all lectures, including this one.
- Readings: Chapters 2, 3 (skip 3.4-6), 4 (skip 4.6, 4.8), 5, 6 (from the textbook)
- Problems:
  - Understanding expressions, programs
  - Debugging
  - Writing code
  - Multiple choice