

Template MetaProgramming

Jim Fawcett

CSE687 – Object Oriented Design
Spring 2005





Overview

- **Template Specializations**
- **Instantiation**
- **Type Deduction Rules**
- **Template Meta-Programs**
- **Functions**
- **Iteration**
- **Selection**
- **Type Lists**



Template Specializations

- A template specialization is the narrowing of the number of types a template accepts as arguments, possibly to one.
 - Template `<class T> widget`
 - generic
 - Template `<class T*> widget`
 - partial specialization to pointers
 - Template `<std::string> widget`
 - full specialization to one concrete type.
 - See `PTS.cpp` in `cse687/code/templates`



Specialization Targets

- All template classes can be specialized
 - `template <class T> class stack {...};`
 - `template<> class stack<int> {...};`
 - Specializations need not provide exactly the same members. You may add or delete members.
 - `template <class U, class V>`
`class Widget {...};`
 - `template <class U, int>`
`class Wiget {...};` // may add or remove
// members to suit int param



Template Instantiation
