

CS 3460

Introduction to Type Inference



Automatic Type Inference

- Types can be inferred from...
 - an initializer
 - return statement
 - function/method argument usage

only discussing inference from initializers right now



Automatic Type Inference – Code Demo



Automatic Type Inference – Inferring Arrays

```
auto primes = { 2, 3, 5, 7 };
```

- You might expect `primes` is inferred as an array
- Instead it is inferred as an `std::initializer_list`
- Because `{ 2, 3, 5, 7 }` is an `std::initializer_list`

```
auto primes = { 2, 3, 5, 7 };  
std::vector primes1 = primes;
```

- To initialize, but not infer an `std::vector` or `std::array`

```
std::vector primes{ 2, 3, 5, 7 };  
std::array primes{ 2, 3, 5, 7 };
```

Dependent Type Inference

- Define the type of a variable based on the type of another
- Use the `decltype` keyword

```
int a{4};  
decltype(a) b{a}; // also decltype(a) b = a;
```

- Other uses to be aware of

```
decltype(auto) c{a}; // c is int, inferred from a  
decltype(auto) d{(a)}; // d is reference to an int
```



Dependent Type Inference – Code Demo

