Programming in C++

Vectors

Curtis Larsen

Utah Tech University—Computing

Fall 2024

Objectives:

- Understand the need for vectors
- Understand functionality of vectors
- Demonstrate use of vectors

Purpose of vectors.

- Store sequential collection of items of same type
- Read items in collection
- Process all items in collection
- Modify items in collection
- Add items to collection
- Remove items from collection

Standard Usage

Standard usage

- `#include <vector>` // to access the vector class
- `std::vector<int> primes;` // to declare a vector of integers
- `primes.resize(10);` // to make room for 10 integers
- `primes[0] = 2;` // to store a value at the first position
- `primes[9] = 29;` // to store a value at the last position
- `int x = primes[0]+primes[9];` // to read values from a vector
- `primes[0] = 3;` // to modify the value at the first position

More Usage

More usage

- `std::vector<std::string> words;` // to declare a vector of strings
- `std::vector<Widget> parts;` // to declare a vector of Widget objects
- `std::vector<double> rational(100);` // to declare a vector of 100 doubles
- `std::vector<double> irrational(100, 3.14);` // to declare a vector of 100 doubles, all initialized to 3.14
- `rational.resize(75);` // to change the size to 75
- `rational.resize(150, 2.5);` // to change the size to 150, all new positions are initialized to 2.5

Processing All Elements

Processing all elements in a vector

```
std::vector<int> primes;
...
for(unsigned int i = 0; i < primes.size(); i++) {
    std::cout << primes[i] << std::endl;
}
```

Processing All Elements

Processing all elements in a vector

```
std::vector<int> primes;
...
for(auto iter = primes.begin(); iter != primes.end(); iter++)
    std::cout << *iter << std::endl;
}
```

More

▸ C++ Reference