Template Functions

Lecture 9
Sections 16.2

Robb T. Koether

Hampden-Sydney College

Fri, Feb 2, 2018

1. Algorithms

2. Template Functions
   - C-Style Templates
   - C++-Style Templates

3. Multiple Instantiation

4. Assignment

Outline

1. Algorithms

2. Template Functions
   - C-Style Templates
   - C++-Style Templates

3. Multiple Instantiation

4. Assignment

Algorithms and Functions

- A function expresses an algorithm.
- Often an algorithm can be applied to many different data types without change.
- For example, the selection sort algorithm will sort a list of objects of any type, provided
  - The operator $<$ is defined.
  - The operator $=$ is defined.
- Is anything else required?

Outline

1. Algorithms

2. Template Functions
   - C-Style Templates
   - C++-Style Templates

3. Multiple Instantiation

4. Assignment

Template Functions

Definition (Template Function)

A template function is a function that contains the code necessary for an algorithm, but does not specify the data types involved. The data types will be specified later, at which point all necessary operators and functions must be defined for those types.

Outline

1. Algorithms

2. Template Functions
   - C-Style Templates
   - C++-Style Templates

3. Multiple Instantiation

4. Assignment

C Template Functions

C Style Template Functions

```
typedef Type T;
void sort(T list[], int size);
```

- Use `T` in place of the type in the function.

Outline

1 Algorithms

2 Template Functions
- C-Style Templates
- C++-Style Templates

3 Multiple Instantiation

4 Assignment

C++ Template Functions

C++ StyleTemplate Functions

```
template <class T>
void sort(T list[], int size);
```

- Write

  ```
  template <class T>
  ```

  at the beginning of the function definition.
- Use T in place of the type in the function.

Outline

1. Algorithms
2. Template Functions
   - C-Style Templates
   - C++-Style Templates
3. Multiple Instantiation
4. Assignment

Multiple Instantiation

- A C++ template function is automatically and separately instantiated for as many types as necessary in a program.
- More than one template type is possible in a single function.
- Give each type a separate name, e.g., `T1`, `T2`, `T3`, etc.

Outline

1. Algorithms
2. Template Functions
   - C-Style Templates
   - C++-Style Templates
3. Multiple Instantiation
4. Assignment

Assignment

Assignment

- Read Section 16.2.