

# CS11600: Introduction to Computer Programming (C++)

## Lecture 15

Svetlozar Nestorov  
*University of Chicago*

## Outline

- Abstract base classes
- Run-time type identification (RTTI)
- Applications of RTTI
- Dynamic casting

2/26/2003

Svetlozar Nestorov, CS 116: Intro to Programming II

2

## Abstract Base Class

- An abstract base class is never instantiated
  - often sits at the top of object hierarchy.
- Pure virtual functions

```
virtual void fun() = 0;
```

- No implementation, derived classes must override it.

2/26/2003

Svetlozar Nestorov, CS 116: Intro to Programming II

3

## Run-Time Type Identification

- RTTI: figure out the type of an object at run-time.
- Why? Polymorphism solves the problem?
  - Not always.
- Several derived classes from the same base have different member functions.

2/26/2003

Svetlozar Nestorov, CS 116: Intro to Programming II

4

## Application of RTTI

- When polymorphism is not appropriate:
  - Extending class libraries without modifying them.
  - Derived classes have member functions not appropriate for other derived classes.

2/26/2003

Svetlozar Nestorov, CS 116: Intro to Programming II

5

## Dynamic Casting

- Language support for RTTI.
  - `dynamic_cast`
    - Works with pointers and references of **polymorphic** classes.
    - Return 0 if unsuccessful for pointers; raises exception for references.
- ```
dynamic_cast<Room *>(loc)
```

2/26/2003

Svetlozar Nestorov, CS 116: Intro to Programming II

6

## Typeid Operator

- Determines the type of an object.
  - Dynamically for expressions of polymorphic types
  - Statically otherwise.
  - A pointer, which points to an object of polymorphic type, **is not** polymorphic!
- Typeid is overloaded for:
  - Type, e.g. `typeid(int)`
  - Expression, e.g. `typeid(loc)`