

Introduction

Overriding functions

Upcasting

Polymorphism

Virtual functions

Abstract classes

Virtual destructors

# Polymorphism

Comp Sci 1575 Data Structures



## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

**Q:** What's the object-oriented way  
to become wealthy?

**A:** Inheritance

## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

### 1 Introduction

Overriding functions

Upcasting

### 2 Polymorphism

Virtual functions

Abstract classes

Virtual destructors

## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

### 1 Introduction

- Overriding functions
- Upcasting

### 2 Polymorphism

- Virtual functions
- Abstract classes
- Virtual destructors

# Overriding functions from a base class

## Introduction

### Overriding functions

### Upcasting

## Polymorphism

### Virtual functions

### Abstract classes

### Virtual destructors

- In a new child class, we can re-define a function first defined in the parent class.
- A function that is redefined must have exactly the same declaration in both base and derived class; that means the same name, same return type, and same parameter list.
- Check out the code

## Introduction

Overriding functions

**Upcasting**

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

### 1 Introduction

Overriding functions

**Upcasting**

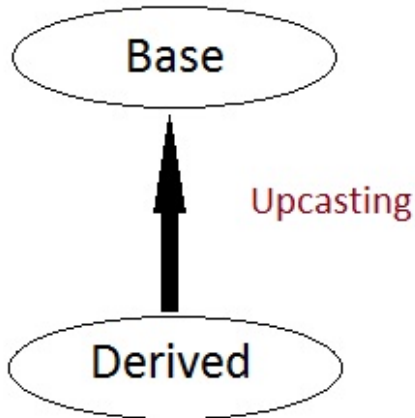
### 2 Polymorphism

Virtual functions

Abstract classes

Virtual destructors

- Up-casting is using a pointer to the Parent class to refer to a Child class object.
- Check out code



## Introduction

Overriding functions

### **Upcasting**

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors



## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

### 1 Introduction

- Overriding functions
- Upcasting

### 2 Polymorphism

- Virtual functions
- Abstract classes
- Virtual destructors

## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

- Polymorphism generally means having many forms
- Up-casting via a pointer to the base class, to call identically defined overridden functions for for each class that redefines and calls that function which was originally in the base class
- Check out code

## Introduction

Overriding functions

Upcasting

## Polymorphism

**Virtual functions**

Abstract classes

Virtual destructors

- 1 Introduction**
  - Overriding functions
  - Upcasting

- 2 Polymorphism**
  - Virtual functions**
  - Abstract classes
  - Virtual destructors

## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

- **virtual** keyword makes a member function of the base class into a function which can be overridden and accessed via a pointer to the base class.
- Only the Base class method's declaration needs the virtual Keyword, not the definition.
- If a function is declared as virtual in the base class, it will be virtual in all its derived classes.

## Introduction

Overriding functions

Upcasting

## Polymorphism

**Virtual functions**

Abstract classes

Virtual destructors

## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

**Abstract classes**

Virtual destructors

- 1 Introduction**
  - Overriding functions
  - Upcasting

- 2 Polymorphism**
  - Virtual functions
  - Abstract classes**
  - Virtual destructors

## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

Virtual destructors

- Contain at least one pure virtual function
- Can have normal functions and variables along with a pure virtual function.
- Classes inheriting an abstract class must implement all pure virtual functions, or else they will become abstract too.
- Mainly used for up-casting, so that derived classes can use interface.
- Can NOT be used to instantiate objects, but pointers and references of abstract class type can be created serving only as an **interface**
- Classes that can be used to instantiate objects are called **concrete classes**
- Check out code

## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

**Virtual destructors**

- 1 Introduction**
  - Overriding functions
  - Upcasting

- 2 Polymorphism**
  - Virtual functions
  - Abstract classes
  - Virtual destructors**



## Introduction

Overriding functions

Upcasting

## Polymorphism

Virtual functions

Abstract classes

**Virtual destructors**

- Virtual destructor is required to avoid memory leaks with dynamically defined polymorphic classes
- Check out code