Introduction

Definition

Examp

ccess contro

protected mo

private mode

Inheritance

Types of inheritance

Single

Hierarchical

Multiple

Multilevel

Not inherit

Constructors

0011361 4060013

class

default

Polymorphism and abstract classes

Inheritance

Comp Sci 1575 Data Structures



Inheritance

Introduction

Definiti

Purpos

Access cont

protected m

Inheritance

Inheritano

Single

Hierar

Multi

Multil

Muhri

Not inheri

Constructors

Constructor of ba

Parameterized vers

Polymorphism and abstract "The great thing about Object Oriented code is that it can make small, simple problems look like large, complex ones."



Introduction

Definition Purpose

Lxample

Access contr

public mode protected mo

Inheritano

IIIIIciiicanic

0) ...

Single

Hierarchi

.

Multile

Hybrid

Not inheri

Constructor

Parameterized versi

Polymorphism and abstract

1 Introduction

Definition Purpose Example

2 Access contro

protected mode private mode

3 Inheritance

Types of inheritance Single

Hierarchical

Multiple

Multileve

Hybrid

Constructor of base class

Parameterized versus defaul



Introduction

Definitio

Purpose

Access contr

public mode protected mod

I m la muita a m a

Inheritanc

Single

Single

Hierarchi

Multiple

Multile

Hybrid

Constructor

Class
Parameterized versi

Polymorphisn and abstract

1 Introduction

Definition

Purpose

Example

2 Access control

protected mode private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multileve

Hybrid

ot inherite

4 Constructors

Constructor of base class
Parameterized versus defau

- When creating a class, rather than writing completely new data members and member functions, make a new class inherit members of an existing class
- Existing class whose properties are inherited by new class is called the Parent, Base, or Super class
- New class which inherits properties of base class is called Child, Derived, or Sub class
- The derived class inherits the members of the base class, on top of which it can add its own members.
- Inheritance is the capability of one class to acquire properties and characteristics from another class



Introduction

Definition

Access contr

protected mode private mode

Inheritance

micritanic

C: 1

Single

Hierarchi

mierarcii

Multiple

IVIUILII

Hybrid

~ . .

Constructor

Parameterized years

Polymorphism and abstract $oldsymbol{1}$ Introduction

Definition

Purpose

Example

2 Access control

protected mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

. Multileve

Hybrid

ot inherite

4 Constructors

Constructor of base class



Purpose of inheritance

Definition
Purpose

Access cont

public mode protected mode private mode

Inheritance

Single Hierarchical Multiple

Multilevel Hybrid

Not inherited

Constructor of base class

Parameterized versus default

Polymorphism and abstract

- Code re-usability: When a new class inherits an existing class, all its methods and fields become available in the new class
- 2 Makes it easier to create and maintain large applications
- 3 Allows abstract classes as interfaces (more coming up soon)



Introduction

Definition

·

Access contr

public mode protected mode private mode

Inhoritono

IIIIeritance

Types of Inn

Single

Hierarchi

mierarciii

Multile

Hybrid

Not inheri

Constructor

Constructor of b

Parameterized vers

Polymorphism and abstract classes

Introduction

Definition

Purpose

Example

2 Access control

public mode protected mod

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multileve

Hybrid

ot inherited

4 Constructors

Constructor of base class
Parameterized versus defau



Base and derived classes

Introduction

Definition

_

Example

Accord contro

public mode

private mod

Inheritance

Types of inheritance

Single

Hierarchical

Multiple

Multileve

пурпа

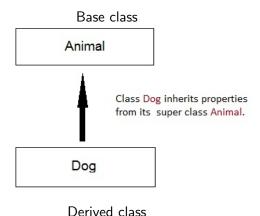
- . .

Constructors

Constructor of bas class

Parameterized versus

Polymorphism and abstract



Class derivation list specifies the base class

```
ntroduction
Definition
Purpose
Example
Access cont
public mode
```

Inheritance

Hierarchical Multiple Multilevel

Constructors

Constructor of base class

Parameterized versus

Polymorphism and abstract

```
class Animal {
  public:
    int brain = 1;
// Class derivation list:
class Dog: public Animal {
  public:
    int spots = 34;
int main() {
  Dog d;
  cout \ll d.brain; //1
  cout << d.spots; // 34
```



Introduction

Definitio Purpose

Access control

Access contr

protected mo

Inheritance

IIIIIeiitaiic

Types of filler

Single

Hierarchi

Multiple

Multile

Hybrid

Constructor

December in all con-

Parameterized versu default

Polymorphism and abstract classes

Introduction

Definition

Example

2 Access control

public mode protected mode private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multilev

Hybrid

ot inherite

4 Constructors

Constructor of base class Parameterized versus defau

Access control
public mode
protected mode

Types of inherit
Single
Hierarchical
Multiple

Not inherited

Constructors

Constructor of base class

Parameterized vers

Polymorphism and abstract

Access levels in the original base class:

- 1 Public: Base class's public members are accessible to all
 - Protected: Base class's protected members are accessible only to the derived class(s)
- 3 Private: Base class's private members are never accessible directly from a derived class, but can be accessed through calls to the public and protected member functions of the base class.

Polymorphism and abstract classes

Modifiers of the original access specifiers:

- Public: public members of the base class become public members of the derived class and protected members of the base class become protected members of the derived class. This is most common.
- Protected: public and protected members of the base class are demoted to protected members of the derived class
- **3 Private**: public and protected members of the base class are demoted to private members of the derived class

Access control modes modify access specifiers

Original access specifier in base:

Who has access?	public	protected	private
members of the same class	yes	yes	yes
members of derived class	yes	yes	no
non members	yes	no	no

Access control

Base class has members that are public, protected, and private (left index)

	Derived Class	Derived Class	Derived Class	
Base class	Public Mode	Protected Mode	Private Mode	
Public	Public	Protected	Private (accessible)	
Protected	Protected	Protected	Private (accessible)	
Private	Not accessible	Not accessible	Not accessible	

Derived class (top indices) inherits these levels (intersections)



Introduction

Definitio

public mode

protected n

. . .

inneritanc

T (1)

Single

Hierarch

Hierarch

Multiple

IVIUILII

Hybri

- . .

Constructor

Parameterized vers

Polymorphism

Polymorphisn and abstract classes

Introduction

Definition

Purpose

Example

2 Access control

public mode

protected mode

private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multileve

Hybrid

ot inherite

4 Constructors

Constructor of base class

```
---
```

```
class Animal {
  string thoughts:
  public:
    int brain = 1;
};
class Dog : public Animal {
  public:
    int spots = 34:
    void printSpots() {cout << brain;} // ??</pre>
int main() {
 Dog d;
  cout << d.brain; // ??
 d.printSpots(); // ??
  cout << d.spots; // 34
  cout << d.thoughts; // ??
```



Definition

Purpose Example

Access contr

protected mo

Inheritance

micritanic

Single

Miorarch

Hierarch

.

Hybri

Not inher

Constructor

class

Parameterized versi default

Polymorphisn and abstract classes

Introduction

Definition

Purpose

Example

2 Access control

public mode

protected mode

private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multileve

Hybrid

ot inherite

4 Constructors

Constructor of base class

Parameterized versus default

```
class Animal {
  public:
    int brain = 1;
};
  public:
    int spots = 34:
};
int main() {
Dog d:
d.printSpots(); ??
```

```
// Class derivation list:
class Dog : protected Animal {
    void printSpots() {cout << brain;} // ??</pre>
 cout << d.brain; // ??
 cout << d.spots; // 34
```



Introduction

Definition

Purpose

Access contr

public mode protected m

protected mo private mode

Inheritanc

IIIICITCATIC

Single

Hierard

Hierarch

Multile

Hybrid

Not inheri

Constructor

Parameterized were

Polymorphism and abstract

Introduction

Definition

Furpose

Example

2 Access control

public mode protected mod

private mode

Inhoritanco

Types of inheritance

Single

Hierarchica

Multiple

. Multileve

Hybrid

ot inherite

4 Constructors

Constructor of base class

```
Computer Science
```

```
private mode
                         };
```

```
class Animal {
  public:
    int brain = 1;
};
// Class derivation list:
class Dog : private Animal {
  public:
    int spots = 34:
    void printSpots() {cout << brain;} // ??</pre>
int main() {
  Dog d:
  cout << d.brain; // ??
  d.printSpots(); ??
  cout << d.spots; // 34
```

Introduction

Dofinit

.

Exami

Access contro

public mode

private mode

Inharitana

Types of fillerin

Single

Hierarchio

Multiple

Multile

Hybrid

....

Constructor of bas

Parameterized versu

Polymorphism

claceae



Introduction

Definitio Purpose

.

Access contr

protected mode private mode

Inheritance

Inheritano

Single

Single

Hierarchio

Multiple

Multil

Hybrid

Not innen

Constructor

class

Parameterized versu default

Polymorphism and abstract classes

Introduction

Definition

Furpose

Example

Access contro

protected mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multile

Hybrid

lot inherit

Constructors

Constructor of base class
Parameterized versus defau



Introduction

Definition Purpose

Accord contr

public mode protected mod

Inheritance

Types of inheritance

Single

Single

Hierarchic

Multiple

Multil

Hybrid

Constructor

Parameterized versu

Polymorphism and abstract

Introduction

Definition

Evample

Example

Access control

public mode protected mod private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multilevel

Hvbrid

t inherite

4 Constructors

Constructor of base class



Types of inheritance: Single

Introduction

Definition

Examp

ess contr

protected mode

Inheritance

.....

Single

Multiple

Multi

Not inhe

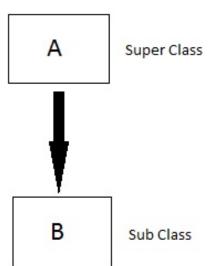
Constructor

Constructor of bas

Parameterized versu default

Polymorphism and abstract

One derived class inherits from only one base class. It is the most simplest form of Inheritance.





Types of inheritance: Hierarchical

Introduction

Definition Purpose

Access contro

public mode protected mo private mode

Inharita

Types of inherita

Types of Illiferita

Hierarchical

.

N. 4. 1. 11

Hybri

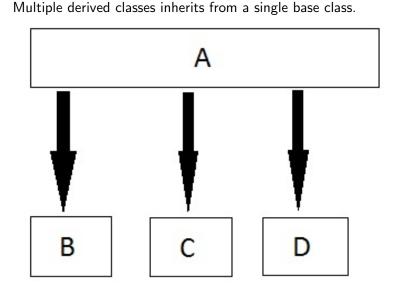
Not innerti

Constructor

class

Parameterized versu default

Polymorphism and abstract



Types of inheritance: Hierarchical

Introduction

Definition

Purpos

public mode

protected m private mod

Inheritano

Types of inheritano

Single

Hierarchical

Multiple

Multi

Hybrid

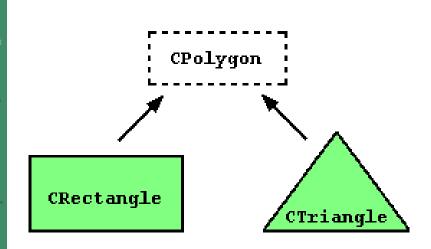
. . .

Constructor

Parameterized year

Parameterized vers default

Polymorphism and abstract



Hierarchical

Types of inheritance: Hierarchical

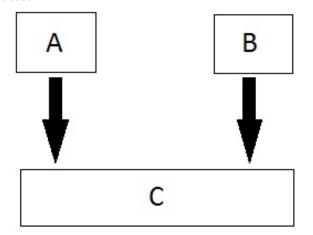
```
class Polygon {
  protected:
    int width, height;
  public:
    void set_values(int a, int b){
      width=a; height=b;
};
class Rectangle: public Polygon {
  public:
    int area(){
      return width * height;
};
class Triangle: public Polygon {
  public:
    int area(){
      return width * height / 2;
};
int main() {
  Rectangle rect;
  Triangle trgl;
  rect.set_values(4.5):
  trgl.set_values(4,5);
  cout << rect.area() << endl; // 20
  cout << trgl.area() << endl; // 10
  return 0:
```



Multiple

Types of inheritance: Multiple

A single derived class may inherit from two or more than two base classes.





Introduction

Definition

public mode

Inheritance

Types of inherita

Single

Hierard

Multiple

Multile

Not inherit

Constructor

class

Parameterized versu default

Polymorphism and abstract classes

Types of inheritance: Multiple

```
class Shape {
   public:
      void setWidth(int w) { width = w; }
      void setHeight(int h) {height = h;}
   protected:
      int width:
      int height;
};
class PaintCost {
   public:
      int getCost(int area) {return area * 70;}
};
class Rectangle: public Shape, public PaintCost {
   public:
      int getArea() {return (width * height);}
};
int main(void) {
   Rectangle Rect:
   int area:
   Rect.setWidth(5);
   Rect. setHeight (7);
   area = Rect.getArea();
   cout << Rect.getArea() << endl;</pre>
   cout << Rect.getCost(area) << endl; // 2450
   return 0:
```



1 ...

Definition Purpose

Access contro

public mode protected mode private mode

Inheritano

Types of inheritan Single Hierarchical

Hybrid

Not inherited

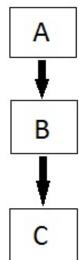
Constructor

Parameterized versu

Polymorphism and abstract

Types of inheritance: Multilevel

Derived class inherits from a class, which in turn inherits from some other class. The Super class for one, is sub class for the other.





Types of inheritance: Hybrid

Introduction

Definition

Evam

Access contro

public mode

1.1. %

Inheritano

Types of i

Single

Multin

Multil

Hybrid

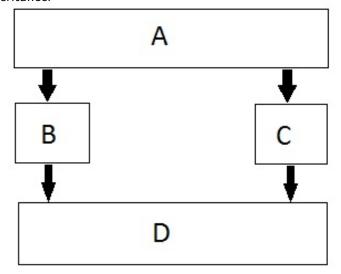
Not innem

Constructors

class

Parameterized versi default

Polymorphism and abstract Hybrid Inheritance is combination of Hierarchical and Mutilevel Inheritance.





Introduction

Definition Purpose

Access contr

public mode protected mode private mode

Inheritance

IIIIIeiitaiic

Single

NA L.

Multil

Hybri

NOC IIIIIEIIC

Constructor

Country of h

Parameterized versi

Polymorphism and abstract classes

Introduction

Definition Purpose

Example

Access control

public mode protected mode private mode

3 Inheritance

Types of inheritance

Single

-lierarchica

Multiple

Multilevel

Hvbrid

Not inherited

4 Constructors

Constructor of base class
Parameterized versus defau



Definition Purnose

.

public mode protected mod

Inheritance

Types of inherit

Hierarchica Multiple

Multilevel

Nine introdu

Not innent

Constructors

Constructor of bas

Parameterized versi default

Polymorphism and abstract

What is not inherited from the base class?

- Constructors, copy constructor, and destructor of the base class
- Assignment operator=()
- Friends of the base class
- Originally private members of the base class (inherited but not accessible)

Introduction

Dofinit

.

Examp

Access contro

public mode protected mo

private mo

Inheritano

Types of inheritant

Single

Hierarchi

Multiple

Multileve

Line and

Not inherited

Construct

Constructor of bas class

Parameterized versus



Introduction

Definitio Purpose

Access contr

protected mode

Inheritance

IIIIIeiitaiic

Single

Hierarch

Multiple

Multile

TTYDIT

Not inherit

Constructors

Constructor of ba

Parameterized versu

Polymorphism and abstract classes

Introduction

Definition Purpose

Example

Access contro

public mode protected mod private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multileve

Hybrid

ot inherite

4 Constructors

Constructor of base class Parameterized versus defau



Introduction

Definition Purpose

public mode

Inheritance

Types of inherita Single Hierarchical Multiple Multilevel Hybrid

Constructors

class
Parameterized versi

Polymorphism and abstract

Constructors of base and derived classes

- Base class constructors are always called with the derived class constructors
- First the base class default constructor is executed, and then the derived class's constructor

Introduction

Definitio Purpose

Accord contr

Access contr

protected mode

Inheritano

IIIICITCATIC

T 01.1

Single

Miorarch

Hierarch

Multiple

....

пурпс

C----

Constructor of base

Parameterized versi

Polymorphism and abstract

Introduction

Definition

Furpose

Example

Access control

public mode

privata mada

private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

. Multileve

Hybrid

ot inherit

4 Constructors

Constructor of base class

Parameterized versus default

```
Computer Scier
```

Constructor of base class is called with the derived

```
Constructor of base
```

```
class Base {
  int x; // Remember, classes default to private
  public:
    Base() {cout << "Base default constructor";}</pre>
};
class Derived : public Base {
  int y;
  public:
    Derived(){cout << "Derived default constr";}</pre>
    Derived (int i){
      y = i;
      cout << "Derived parameterized";</pre>
int main() {
Base b; // Base...
 Derived d1; // Base...
 Derived d2(10); // Base...
```

Introduction

Definitio Purpose

.

Access contr

protected mode

Inheritance

IIIIeritance

Types of inhe

Single

Hierarch

Multiple

Multile

Hybric

Constructo

Parameterized versu

Polymorphism and abstract

Introduction

Definition

Evample

Example

Access control

public mode protected mod

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

. Multileve

Hybrid

Jot inherite

4 Constructors

Constructor of base class

Parameterized versus default



Parameterized versus default

Introduction

Definition Purpose

Accord cont

public mode protected mod private mode

Inheritance Types of inherita

Single
Hierarchical
Multiple
Multilevel

Not inherited

Constructor of ba

Parameterized vers default

- Unless otherwise specified, the constructors of a derived class calls the default constructor of its base class(es)
- To call base class's parameterized constructor inside derived class's parameterized constructor, we must mention it explicitly while declaring derived class's parameterized constructor.



Force calling of base parameterized constructor

```
Single
Hierarchical
Multiple
Multiple
Hybrid
Not inherited
Constructors
Constructor of backas
Parameterized verdefault
Polymorphist
and abstract
classes
```

```
class Base{
 public:
    int x:
    Base(int i) { x = i; }
};
class Derived: public Base {
 public:
   int y;
   // Specify base parameterized constructor
    Derived (int i): Base(i) { // Options here?
     y = i;
int main(){
 Derived d(10);
 cout << d.x; // 10
 cout << d.y; // 10
```

Introduction

Definition Purpose

Access contr

public mode protected mode

Inheritano

IIIIIeritanic

Single

Literana bi

Hierarchi

Multiple

IVIUILII

Trybrid

Constructor

Class
Parameterized versu

Parameterized versus default

Polymorphism and abstract classes

Introduction

Definition Purpose

Example

Access contro

protected mode private mode

3 Inheritance

Types of inheritance

Single

Hierarchica

Multiple

Multilev

Hybrid

100 1111101

Constructors

Constructor of base class

Introduction

Definit

Durno

Examp

ccess contr

public mode protected mod

private mod

Inheritano

.....

Types of inheritance

Hiorard

Hierard

Multip

Multi

Hybric

Not inheri

Constructor

class

Parameterized versi

Polymorphism and abstract classes