Multilevel Inheritance Examples in C#

# Example 1: Educational Institution System

Hierarchy: Person → Teacher → Professor

Code Example:

class Person  
{  
 public string Name;  
 public void ShowName() => Console.WriteLine("Name: " + Name);  
}  
  
class Teacher : Person  
{  
 public string Subject;  
 public void Teach() => Console.WriteLine($"{Name} teaches {Subject}");  
}  
  
class Professor : Teacher  
{  
 public void ConductResearch() => Console.WriteLine($"{Name} is conducting research in {Subject}");  
}  
  
Professor prof = new Professor { Name = "Dr. Smith", Subject = "Physics" };  
prof.ShowName();  
prof.Teach();  
prof.ConductResearch();

# Example 2: Banking System

Hierarchy: Account → SavingsAccount → PremiumSavingsAccount

Code Example:

class Account  
{  
 public double Balance;  
 public void Deposit(double amount) => Balance += amount;  
}  
  
class SavingsAccount : Account  
{  
 public double InterestRate = 0.03;  
 public void AddInterest() => Balance += Balance \* InterestRate;  
}  
  
class PremiumSavingsAccount : SavingsAccount  
{  
 public double BonusRate = 0.02;  
 public void AddBonus() => Balance += Balance \* BonusRate;  
}

# Example 3: Animal Kingdom

Hierarchy: Animal → Mammal → Human

Code Example:

class Animal  
{  
 public void Breathe() => Console.WriteLine("Breathing...");  
}  
  
class Mammal : Animal  
{  
 public void FeedMilk() => Console.WriteLine("Feeding milk...");  
}  
  
class Human : Mammal  
{  
 public void Speak() => Console.WriteLine("Speaking...");  
}

# Example 4: E-Commerce Order System

Hierarchy: Product → ElectronicProduct → Smartphone

Code Example:

class Product  
{  
 public string Name;  
 public double Price;  
}  
  
class ElectronicProduct : Product  
{  
 public int WarrantyYears;  
}  
  
class Smartphone : ElectronicProduct  
{  
 public string OS;  
 public void DisplayDetails()  
 {  
 Console.WriteLine($"{Name} - {OS}, Price: {Price}, Warranty: {WarrantyYears} year(s)");  
 }  
}

# Example 5: Employee Hierarchy

Hierarchy: Employee → Manager → Director

Code Example:

class Employee  
{  
 public string Name;  
 public void Work() => Console.WriteLine($"{Name} is working");  
}  
  
class Manager : Employee  
{  
 public void ManageTeam() => Console.WriteLine($"{Name} is managing a team");  
}  
  
class Director : Manager  
{  
 public void MakeDecisions() => Console.WriteLine($"{Name} is making strategic decisions");  
}