[INHERITANCE]

Course Name: Object Oriented Programming
Course Code: CSE 121
Intake – 50(06)

Presented By –

Name: Asifur Rahman [22234103353]

Name: Ferdouse Hassan Nowrin [22234103237]

Name: Tasnimul Haque Nahin [22234103180]

Name: Mahajabin Kabir Arin [22234103213]

Name: Delwar Hossain Roman [22234103233]

Presented To –

Khan Md. Hasib

Assistant Professor

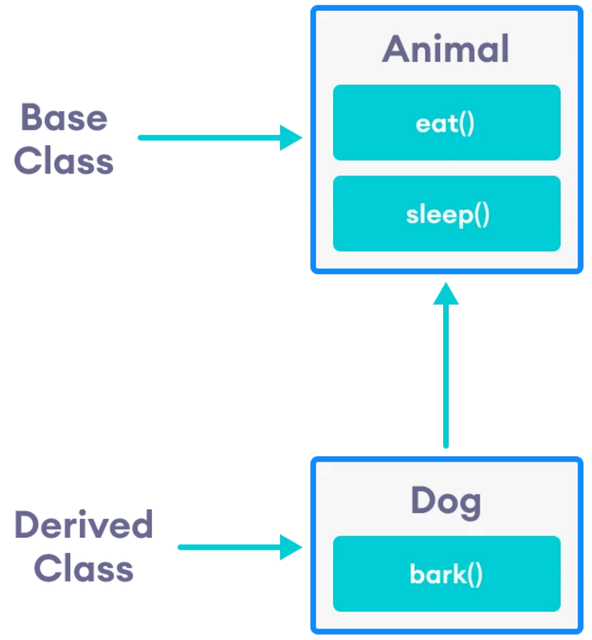
Department of CSE

Bangladesh University of Business and Technology (BUBT)

Email: khanmdhasib@bubt.edu.bd

What is Inheritance?

- Inheritance is a concept in object-oriented programming (OOP) that allows a new class to be based on an existing class, inheriting all of its properties and methods.



[1]

History of Inheritance

- Ole-Johan Dahl and Kristen Nygaard first time included a form of inheritance called subclassing in **Simula** Programming Language. [2]
- Early object-oriented programming language, **Smalltalk** introduced a more sophisticated form of inheritance called class inheritance. [3]
- In C++, inheritance was first introduced in the 1980s as a way to extend the capabilities of C
- **Bjarne Stroustrup** based the inheritance mechanism in C++ on the class inheritance mechanism in Smalltalk. [4]

Syntax of Inheritance in C++

```
class Base {
  // Base class members
};

class Derived : accessSpecifier Base {
  // Derived class members
};
```

Modes of Inheritance

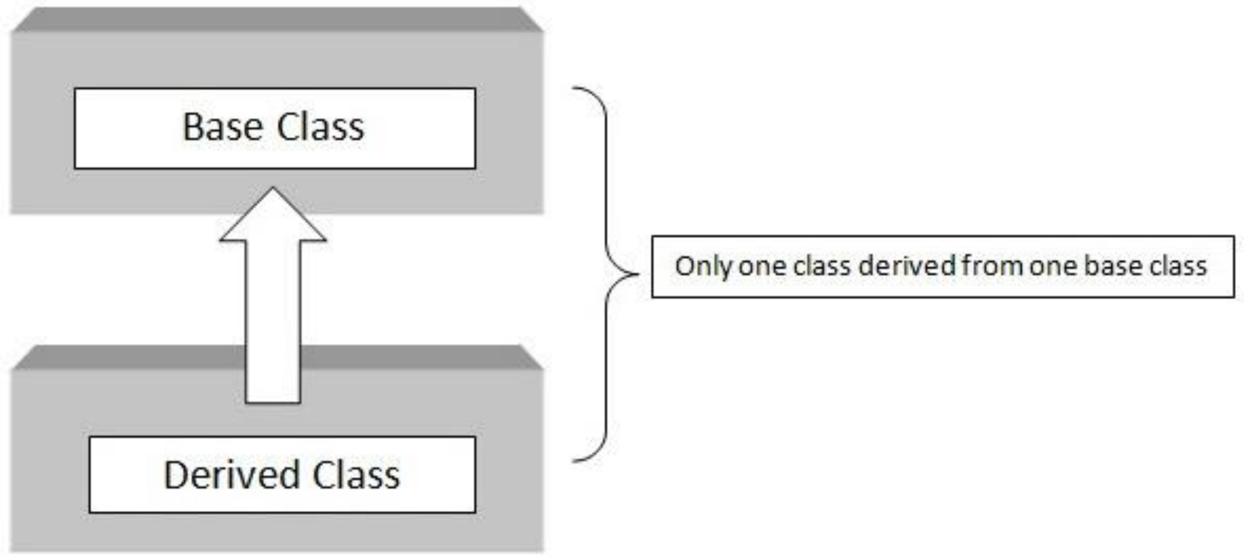
- **Public Mode :** All public members of the base class are accessible in the derived class as public members.
- **Protected Mode :** All public and protected members of the base class become private members of the derived class.
- **Private Mode :** All public and protected members of the base class become protected members of the derived class.

Types of Inheritance

- Single Inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Multilevel Inheritance
- Hybrid Inheritance

Single Inheritance

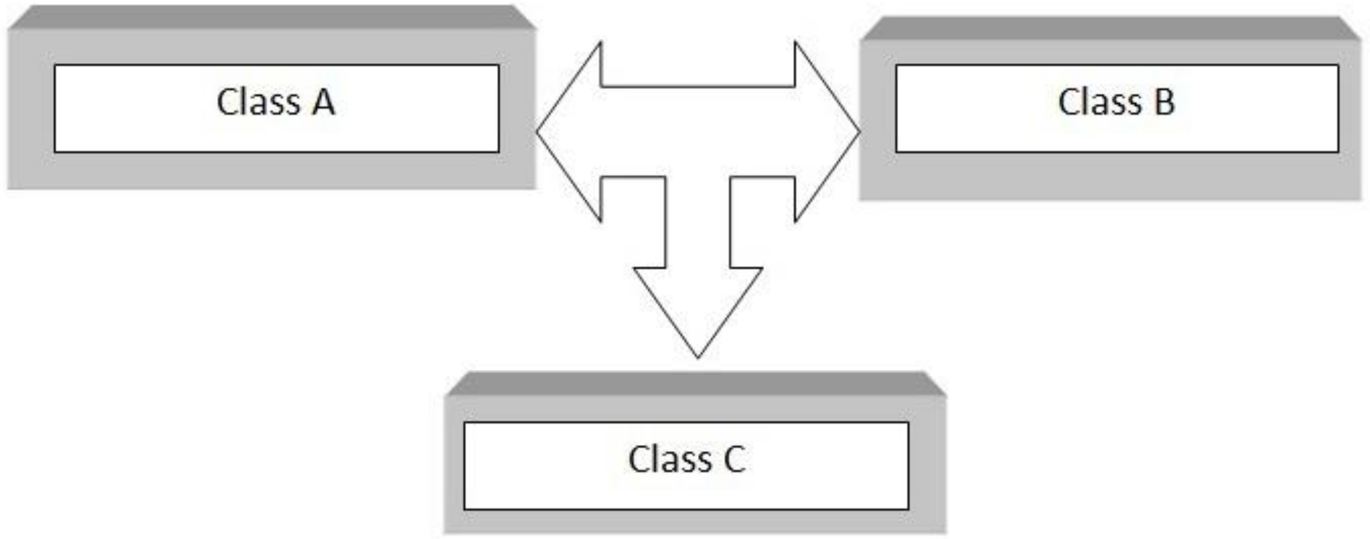
- A derived class is derived from **a single base class**.



[5]

Multiple Inheritance

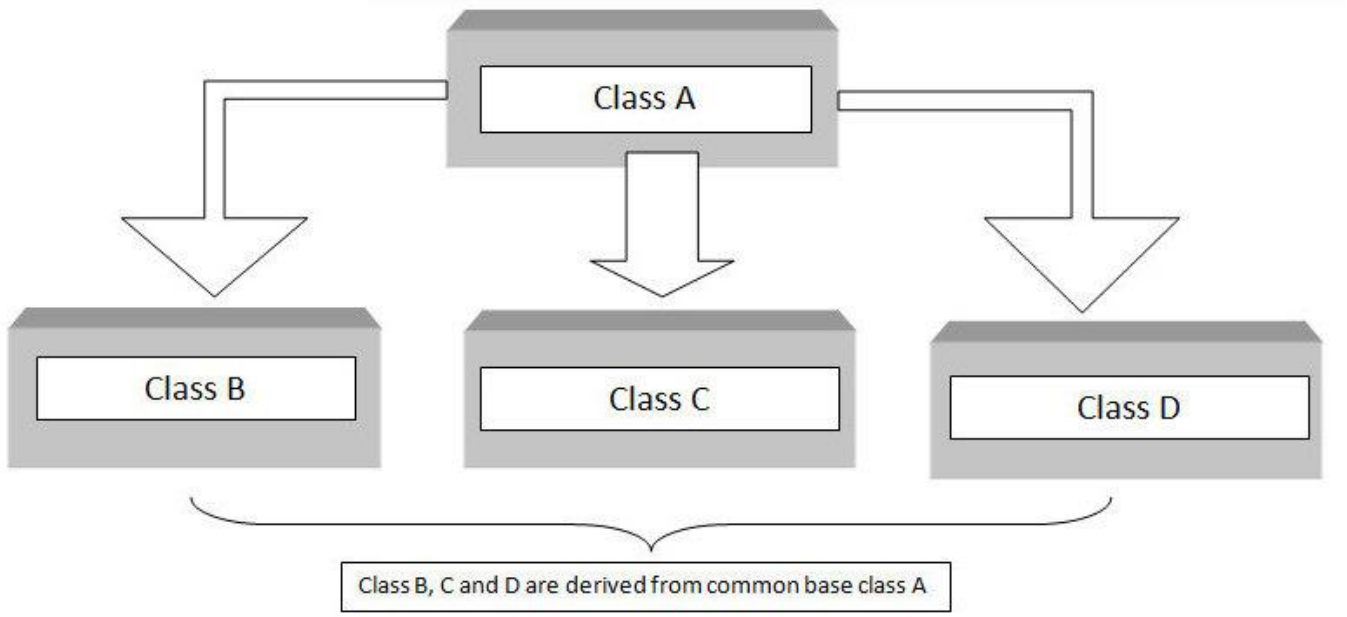
- A derived class is derived from **more than one base class**.



[6]

Hierarchical Inheritance

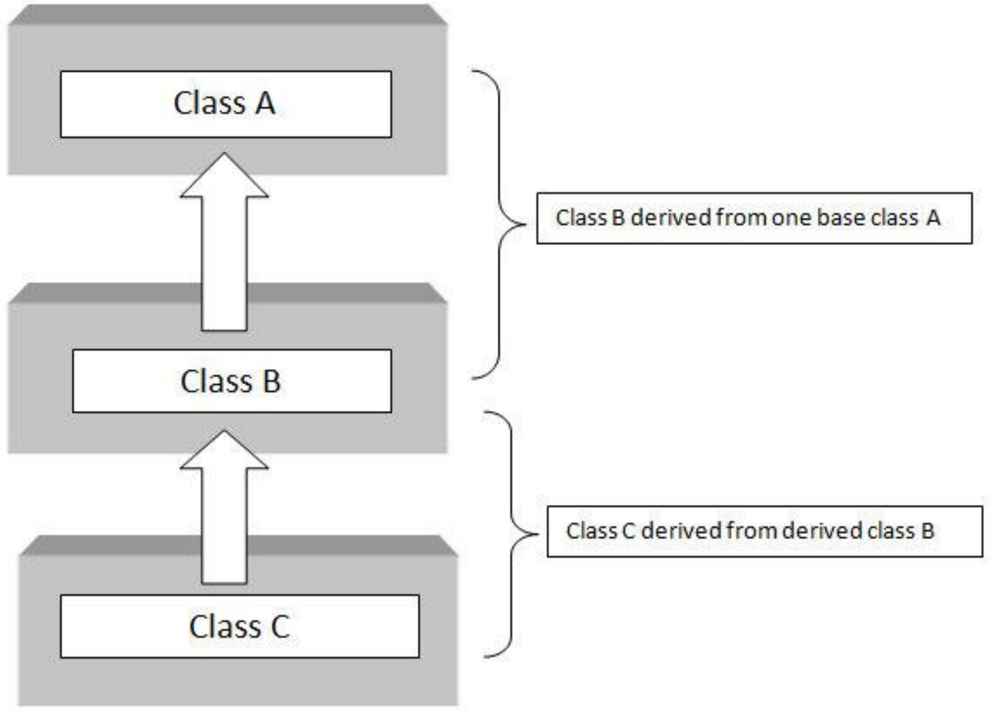
- **Multiple derived classes** are created from a single base class.



[7]

Multilevel Inheritance

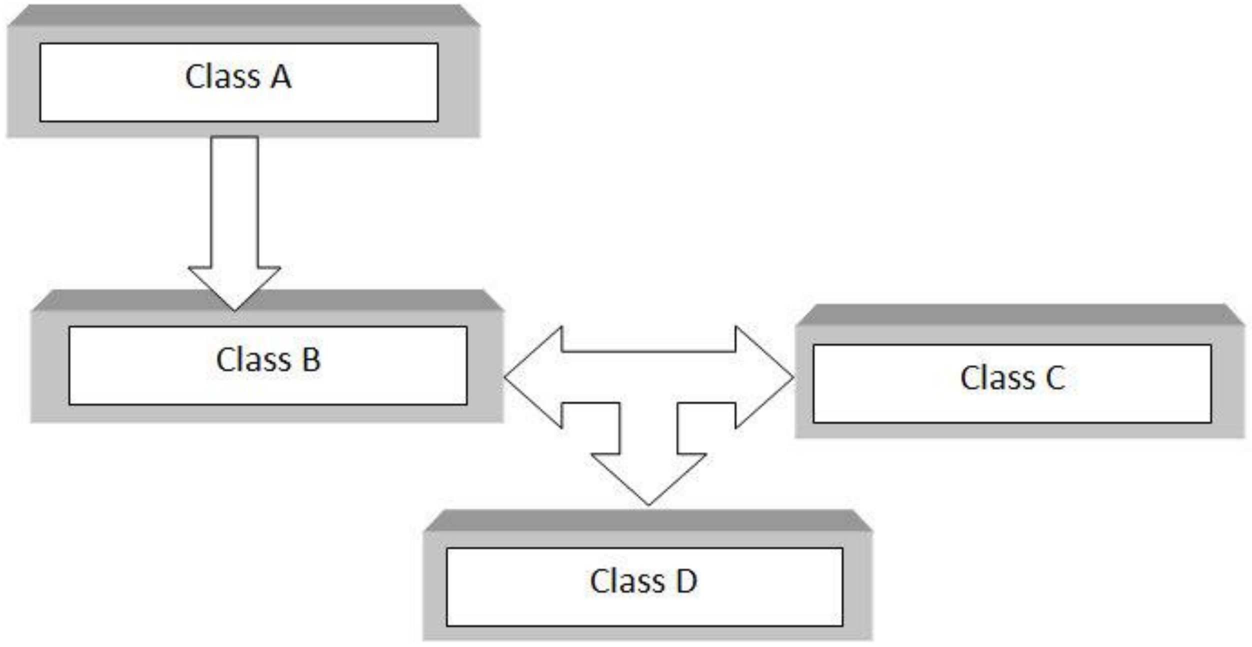
- A derived class is derived from **another derived class**.


Class A
Class B derived from one base class A
Class B
Class C derived from derived class B
Class C

[8]

Hybrid Inheritance

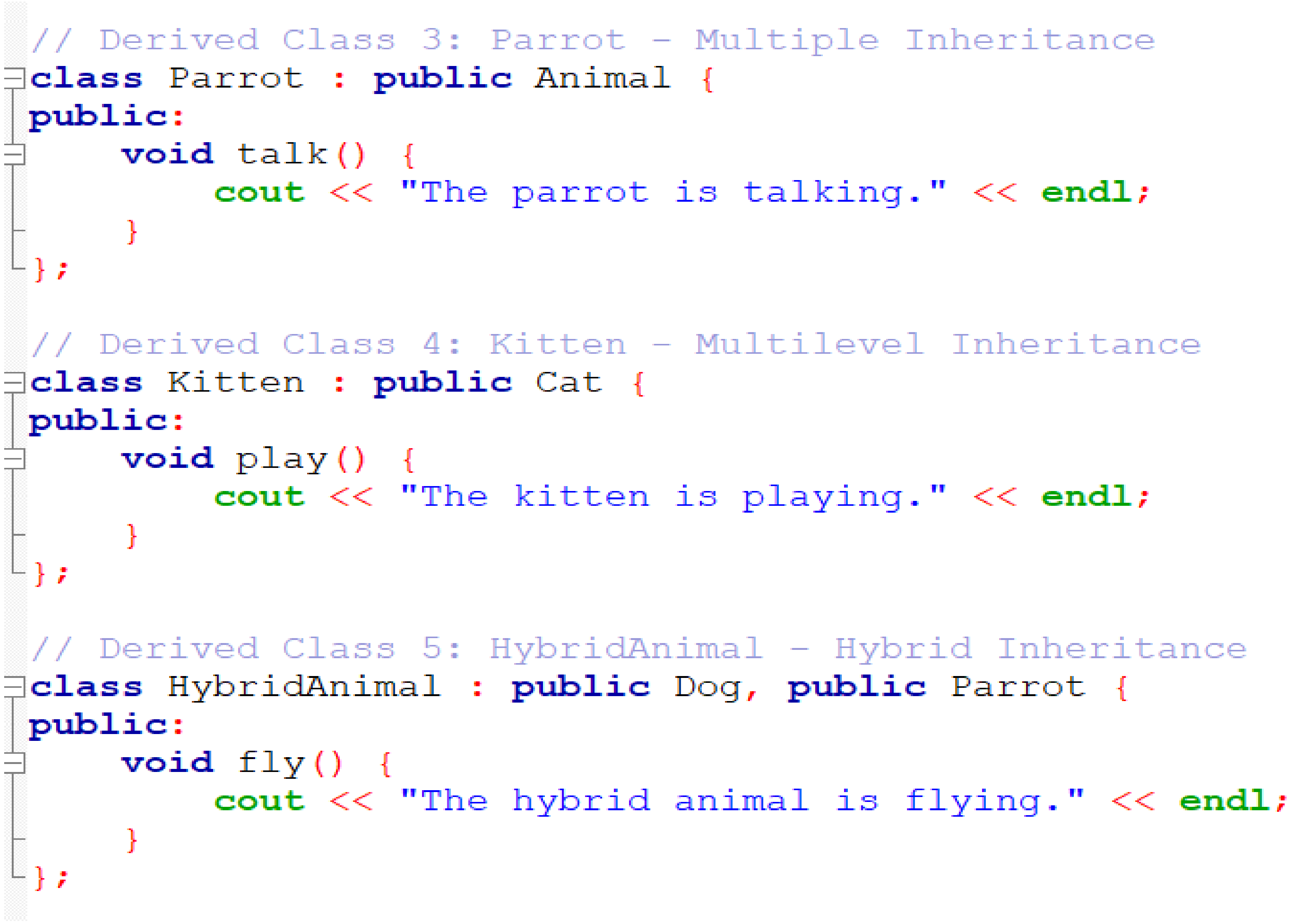
- A combination of multiple types of inheritance, such as a combination of Hierarchical and Multiple or Hierarchical and Multilevel.



[9]

Sample Code Featuring All types of Inheritance

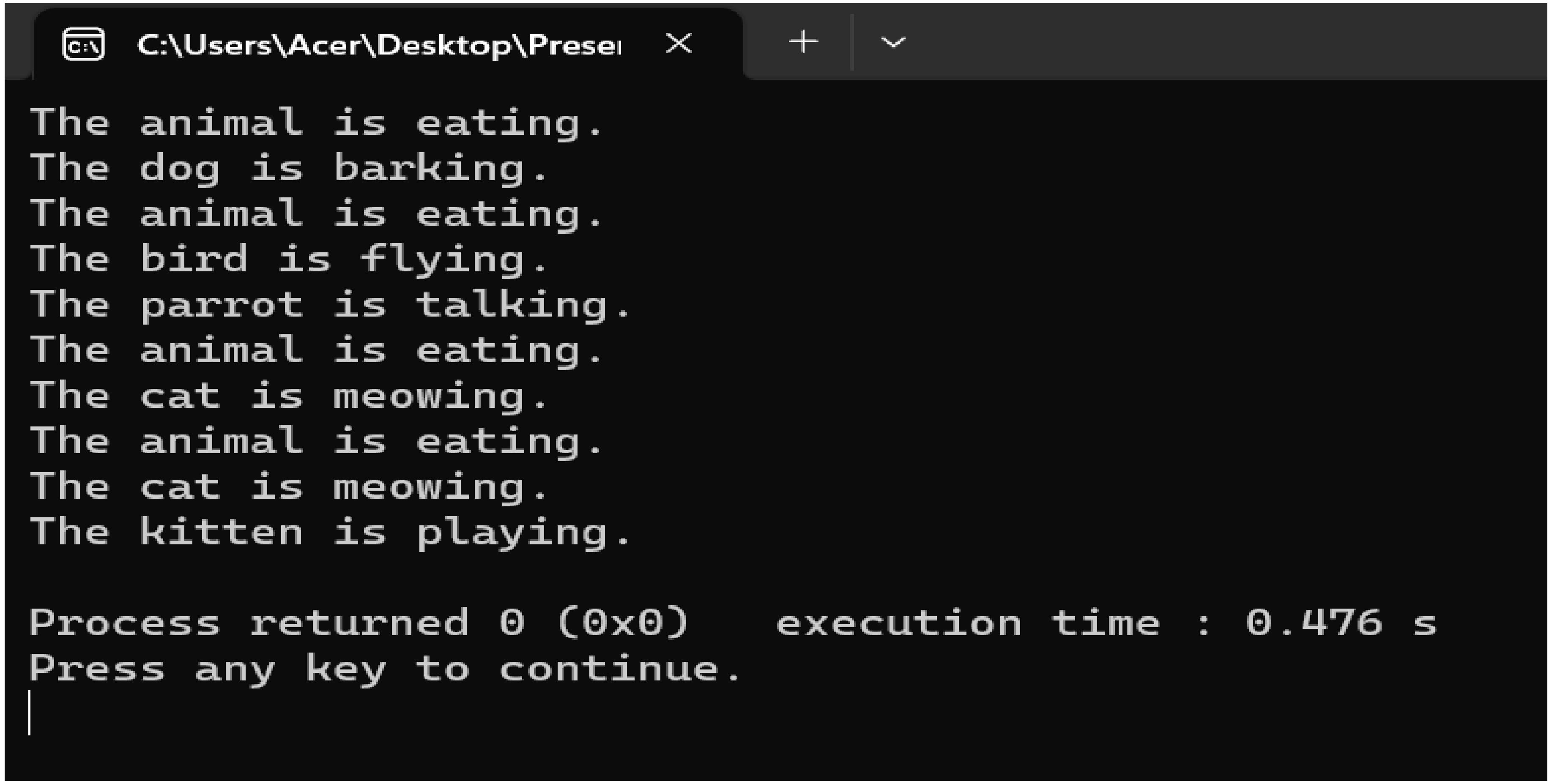
```
#include <iostream>
using namespace std;

// Base Class
class Animal {
public:
    void eat() {
        cout << "The animal is eating." << endl;
    }
};

// Derived Class 1: Dog - Single Inheritance
class Dog : public Animal {
public:
    void bark() {
        cout << "The dog is barking." << endl;
    }
};

// Derived Class 2: Cat - Hierarchical Inheritance
class Cat : public Animal {
public:
    void meow() {
        cout << "The cat is meowing." << endl;
    }
};
```



```cpp
// Derived Class 3: Parrot - Multiple Inheritance
class Parrot : public Animal {
public:
    void talk() {
        cout << "The parrot is talking." << endl;
    }
};

// Derived Class 4: Kitten - Multilevel Inheritance
class Kitten : public Cat {
public:
    void play() {
        cout << "The kitten is playing." << endl;
    }
};

// Derived Class 5: HybridAnimal - Hybrid Inheritance
class HybridAnimal : public Dog, public Parrot {
public:
    void fly() {
        cout << "The hybrid animal is flying." << endl;
    }
};
```

```cpp
int main() {
    // Single Inheritance Example
    Dog d;
    d.eat();
    d.bark();

    // Hierarchical Inheritance Example
    Cat c;
    c.eat();
    c.meow();

    // Multiple Inheritance Example
    Parrot p;
    p.eat();
    p.talk();

    // Multilevel Inheritance Example
    Kitten k;
    k.eat();
    k.meow();
    k.play();

    // Hybrid Inheritance Example
    HybridAnimal ha;
    ha.eat();
    ha.bark();
    ha.talk();
    ha.fly();

    return 0;
}
```

Output



```
The animal is eating.
The dog is barking.
The animal is eating.
The bird is flying.
The parrot is talking.
The animal is eating.
The cat is meowing.
The animal is eating.
The cat is meowing.
The kitten is playing.

Process returned 0 (0x0)   execution time : 0.476 s
Press any key to continue.
```

Benefits of Inheritance

- Reusability
- Code Organization
- Polymorphism
- Code Flexibility

Drawbacks of Inheritance

- Tight coupling
- Complex hierarchies
- Overuse

Reference

1. Programiz - C++ Inheritance
2. Wikipedia - Inheritance (object-oriented programming)
3. Wikipedia - Smalltalk
4. Wikipedia - C++
5. Trytoprogram - C++ Single Inheritance
6. Trytoprogram - C++ Multiple Inheritance
7. Trytoprogram - C++ Hierarchical Inheritance
8. Trytoprogram - C++ Multilevel Inheritance
9. Trytoprogram - C++ Hybrid Inheritance

Thank You All