

Understanding Polymorphism in Programming

1. What is Polymorphism?

Polymorphism is a fundamental concept in object-oriented programming (OOP) that allows objects of different classes to be treated as objects of a common superclass. It provides a way to perform a single action in different forms. Polymorphism promotes flexibility and integration in code reuse and system scalability.

2. Polymorphism in Python

Python Example:

```
class Animal:
    def speak(self):
        print("Animal speaks")
```

```
class Dog(Animal):
    def speak(self):
        print("Dog barks")
```

```
class Cat(Animal):
    def speak(self):
        print("Cat meows")
```

```
def make_animal_speak(animal):
    animal.speak()
```

```
make_animal_speak(Dog())
make_animal_speak(Cat())
```

3. Polymorphism in C++

C++ Example:

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

```
class Animal {
public:
```

Understanding Polymorphism in Programming

```
virtual void speak() {  
    cout << "Animal speaks" << endl;  
}  
};
```

```
class Dog : public Animal {  
public:  
    void speak() override {  
        cout << "Dog barks" << endl;  
    }  
};
```

```
class Cat : public Animal {  
public:  
    void speak() override {  
        cout << "Cat meows" << endl;  
    }  
};
```

```
void makeAnimalSpeak(Animal* animal) {  
    animal->speak();  
}
```

```
int main() {  
    Dog d;  
    Cat c;  
    makeAnimalSpeak(&d);  
    makeAnimalSpeak(&c);  
    return 0;  
}
```