C++ Polymorphism

Polymorphism

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Like we specified in the previous chapter; **Inheritance** lets us inherit attributes and methods from another class. **Polymorphism** uses those methods to perform different tasks. This allows us to perform a single action in different ways.

For example, think of a base class called Animal that has a method called animalSound(). Derived classes of Animals could be Pigs, Cats, Dogs, Birds - And they also have their own implementation of an animal sound (the pig oinks, and the cat meows, etc.):

Example

```
// Base class
class Animal {
 public:
 void animalSound() {
 cout << "The animal makes a sound \n" ;
 }
};

// Derived class
class Pig : public Animal {
 public:
 void animalSound() {
 cout << "The pig says: wee wee \n" ;
 }
};

// Derived class
class Dog : public Animal {
 public:
 void animalSound() {
 cout << "The dog says: bow wow \n" ;
 }
};
```

Remember from the Inheritance chapter that we use the : symbol to inherit from a class.

Now we can create Pig and Dog objects and override the animalSound() method:

Example

```
// Base class
class Animal {
 public:
 void animalSound() {
 cout << "The animal makes a sound \n" ;
 }
};

// Derived class
class Pig : public Animal {
 public:
 void animalSound() {
 cout << "The pig says: wee wee \n" ;
 }
};

// Derived class
class Dog : public Animal {
 public:
 void animalSound() {
 cout << "The dog says: bow wow \n" ;
 }
};

int main() {
 Animal myAnimal;
 Pig myPig;
 Dog myDog;

 myAnimal.animalSound();
 myPig.animalSound();
 myDog.animalSound();
 return 0;
}
```

Why And When To Use "Inheritance" and "Polymorphism"?

- It is useful for code reusability: reuse attributes and methods of an existing class when you create a new class.