California State University
SAN MARCOS

CS 211 Class Inheritance

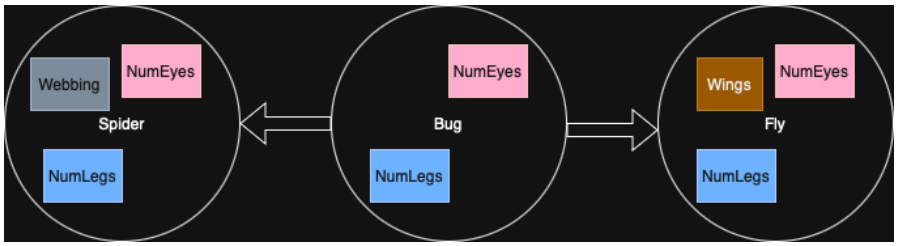
6/25/2025
David E.

1 Classes

A class is a blueprint for creating **objects** that have **properties** (variables) and **methods** (functions). For example, a car could be a class. There are many different cars, but all follow **the same general design. The properties** of a car could be its paint, it's engine, or even the type of seats. **The methods** of a car could be the gas pedal that triggers the engine or the button that activates the hazard lights.

2 Inheritance

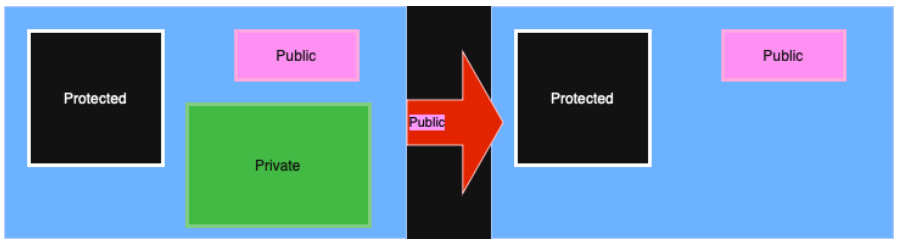
A class can be created **using another class as a base**, which will allow the **new class to have similar properties and methods**. Continuing the car analogy, a car as a concept could be a class with doors and a windshield, so electric cars and engine powered cars could both be inherited classes that have the **original properties** of a car also having a different power source **property**. This is called class inheritance. There are **three types** of class inheritance.



In the diagram above there is a bug class with NumEyes and NumLegs as properties. Spider and Fly are both inherited classes so the maintain the properties of a bug but have their own unique properties as well.

2.1 Public Inheritance

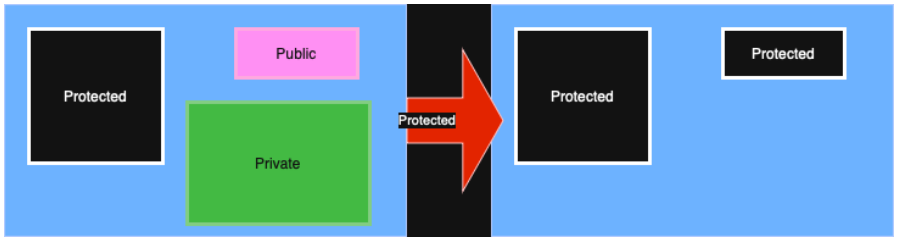
An inherited class that uses the "public" key word will inherit all protected and public properties and methods from the parent class. Public properties and methods **can be changed and accessed anywhere in the program** (in main or in functions) and will always be inherited.



The diagram above depicts inheriting a class using the public key word. As shown only the protected and public variables are inherited and they remain the types they were in the parent class.

2.2 Protected Inheritance

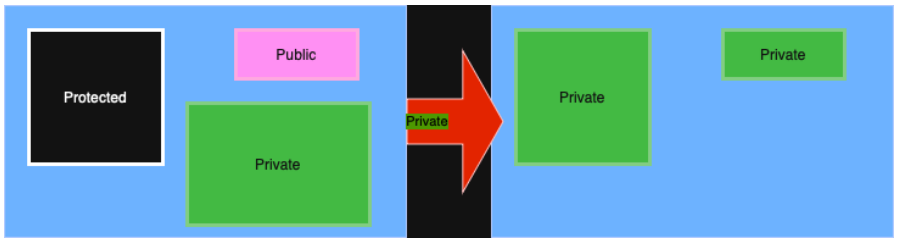
An inherited class that uses the "protected" key word will inherit all protected and public properties and methods from the parent class, but it will make all of the public properties and methods **into protected** properties and methods. Protected properties and methods **can only be changed and accessed within the class's methods**, and **can still be inherited.**



The diagram above depicts inheriting a class using the protected key word. As shown only the protected and public variables are inherited and the public property became protected.

2.3 Private Inheritance

An inherited class that uses the "private" key word will inherit all protected and public properties and methods from the parent class, but it will make all of the public/protected properties and methods **into private** properties and methods. Private properties and methods **can only be changed and accessed within the class's methods** and **cannot be inherited**.



The diagram above depicts inheriting a class using the private key word. As shown only the protected and public variables are inherited and both became private.