Suppose we create a class, Person, that possesses a header file as such.

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

class Person {
public:
   string name;
   string bloodType;

   Person(string name, double bankBalance, string bloodType);
   string getName();
   string getBloodType();
   double getBankBalance();

private:
   double bankBalance;
};
```

Followed by an implementation such as this

```
#include <cstdlib>
#include "Person.h"

using namespace std;
   Person::Person(string name, double bankBalance, string bloodType){
      this->name = name;
      this->bankBalance = bankBalance;
      this->bloodType = bloodType;
   }
   string  Person::getName(){ return name;}
   string  Person::getBloodType() {return bloodType;}
   double  Person::getBankBalance() {return bankBalance;}
```

This will provide a base class for our derived class, Samurai. A Samurai is a Person, so the relation holds in a conventional manner. The derived class is implemented as so

```
#include <cstdlib>
#include <string>
```

```
#include <iostream>
#include "Person.h"
using namespace std;

class Samurai : public Person{
   string swordType;
   string armorType;
public:
   Samurai(string sword, string armor, string Name, string bloodType, double bankBalance):
Person(Name, bankBalance, bloodType){
      swordType = sword;
      armorType = armor;
   }
   string getSword(){
      return swordType;
   }
   string getArmor(){
      return armorType;
   }


};
```

We can test our derived class with the following Main method. Which outputs according to the comments.

```
int main(int argc, char** argv) {
   //We first instigate the base class Person to show it's capabilities as an independent class
   Person person1 = Person("Randolph", 100, "A+");
   //Prints "100"
   cout << person1.getBankBalance() << endl;
   //Prints "Randolph"
   cout << person1.getName() << endl;
   //Prints “A+”
   cout << person1.getBloodType() << endl;

   //Now we examine and test the derived class "Samurai" which is derived from Person
```

```
    //Our new constructor for our derived class takes two additional Parameters: Sword and
Armor
    Samurai samurai1 = Samurai("Jack's Katana", "Robe","Samurai Jack", "O+", 500);

    //Now we can show that it can utilize inherited methods along with its new methods

    //Prints "Robe"
    cout << samurai1.getArmor() << endl;
    //Prints "Jack's Katana"
    cout << samurai1.getSword() << endl;
    //Prints "500"
    cout << samurai1.getBankBalance() << endl;
    //Prints "Samurai Jack
    cout << samurai1.getName() << endl;
    //Prints "O+"
    cout << samurai1.getBloodType() << endl;

}
```

We provide another example, where Samurai is the base class, and Bushido is the derived class. A Bushido is a Samurai, so the relationship holds.

```
#include <cstdlib>
#include "Samurai.cpp"

using namespace std;
class Bushido : public Samurai{
    string rank;
    string landAmount;
public:
    Bushido(string rank, string landAmount,string sword, string armor, string Name, string
bloodType, double bankBalance): Samurai(sword, armor, Name, bloodType, bankBalance){
        this->rank = rank;
        this->landAmount = landAmount;
    }
    string getRank(){
        return rank;
    }
    string getLandAmount(){
```

```
        return landAmount;
    }


};

int main(int argc, char** argv) {
    //We will show a second generation derived class as our second example
    //One that is derived from our first derived class, Samurai

    //We create a Bushido object
    Bushido bushido1 = Bushido("Chief Officer", "100 acres", "Master Blade", "Clay Chieftan
Armor", "George", "B+", 500);
    //We will observe in this example that we can use methods from our base class, Samurai,
and its base Class, Person

    //First, Samurai methods at work

    //Prints "Clay Chieftan Armor"
    cout << bushido1.getArmor() << endl;
    //Prints "Master Blade"
    cout << bushido1.getSword() << endl;

    //Now, let's show some "Person" methods

    //Prints "George"
     cout << bushido1.getName() << endl;
    //Prints "B+"
     cout << bushido1.getBloodType() <<endl;

     //And finally, some Bushido methods

     //Prints "Chief Officer"
     cout << bushido1.getRank() <<endl;

     //Prints "100 Acres"
     cout << bushido1.getLandAmount() <<endl;
};
```