Inheritance

- the _____ (derived/base) class is the _______(parent/child)
- the _____ (derived/base) class is the _______(parent/child)
- a _______ (parent/child) is a ___________ (parent/child)

(More) Concretely

- the _______ class is the _______
- the _______ class is the _______
- a _______ is a ___________

What is not inherited?

What is inherited?

How does privacy interact with inheritance?

What is dynamic dispatch? How does it relate to the `virtual` keyword?

Animal

```
class Animal {
public:
    Animal(string sound): sound_(sound) {}
    string MakeSound() {return sound_; }
    virtual int GetPower() {return 0; }
private:
    string sound_;
}
```

Reptile

```
class Reptile : public Animal {
public:
     Reptile(string sound):
     Animal("rawr") {}
     int GetPower() {return 2; }
}
```

Mammal

```
class Mammal : public Animal {
public:
     Mammal(string sound):
     Animal("fuzzy fuzz") {}
     int GetPower() {return 3; }
}
```

Turtle

```
class Turtle : public Reptile {
public:
     Turtle(string sound):
     Reptile("turtle turtle") {}
     int GetPower() {return 7; }
}
```

```
Turtle t;
Mammal m;
Animal * a = new Turtle()

// which method is being called for these function calls?
std::cout << t.MakeSound() << std::endl;
std::cout << m.MakeSound() << std::endl;
std::cout << a->MakeSound() << std::endl;


// what about for these ones?
std::cout << t.GetPower() << std::endl;
std::cout << m.GetPower() << std::endl;
std::cout << a->GetPower() << std::endl;
```