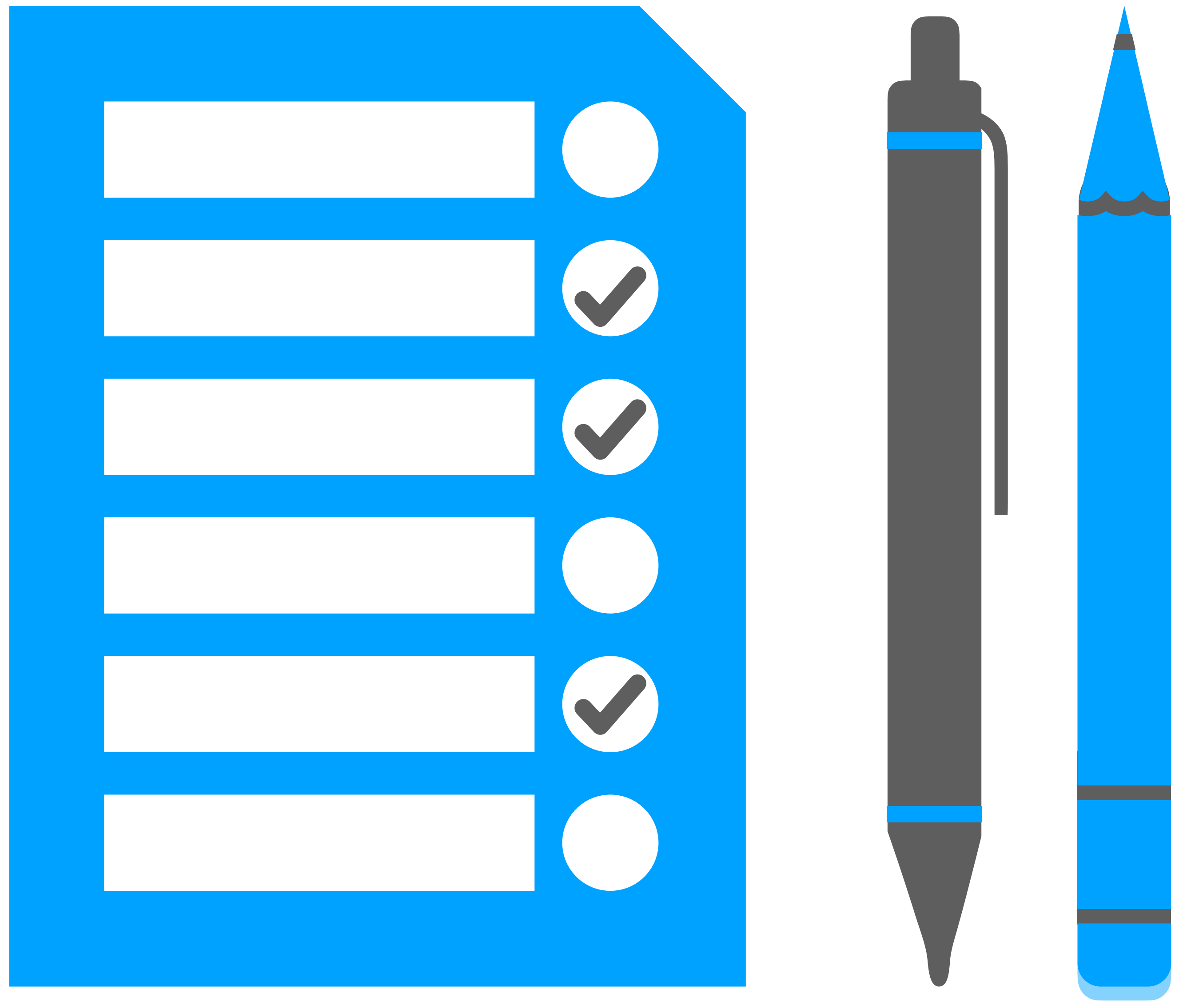
Polymorphism



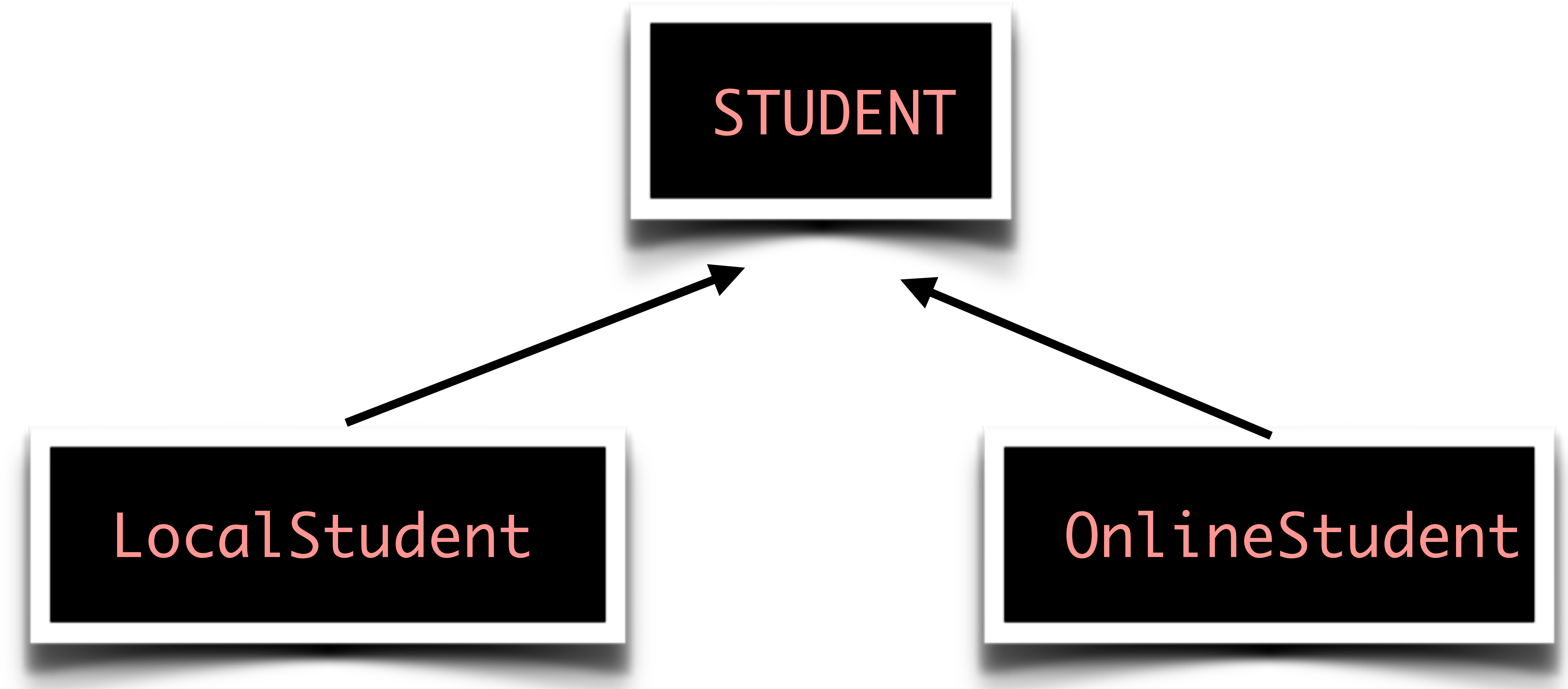
- Review IS-A Relationship
- Understand the concept of polymorphism
- Review Reference Type and Object Type
- Declare and instantiate object in polymorphic way
- Understand the benefit of polymorphism

IS-A Relationship

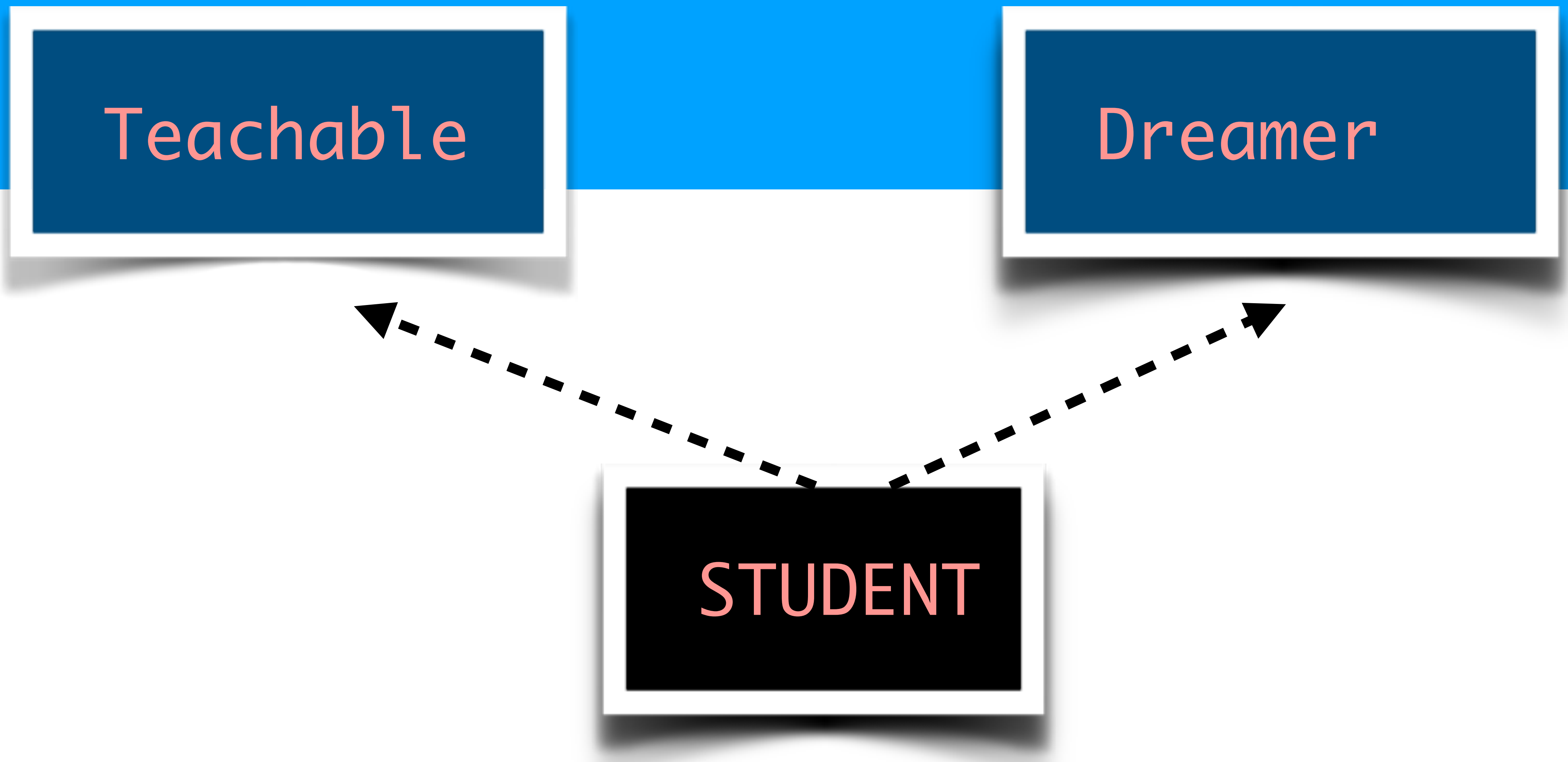
```java
public class Student { // valid code here }
```

```java
public class LocalStudent extends Student { // valid code }
```

```java
public class OnlineStudent extends Student{ //valid code here }
```



IS-A Relationship



```
public class Student implements Teachable, Dreamer { // valid code here }
```

```
public interface Teachable { // valid code }
```

```
public interface Dreamer { //valid code here }
```

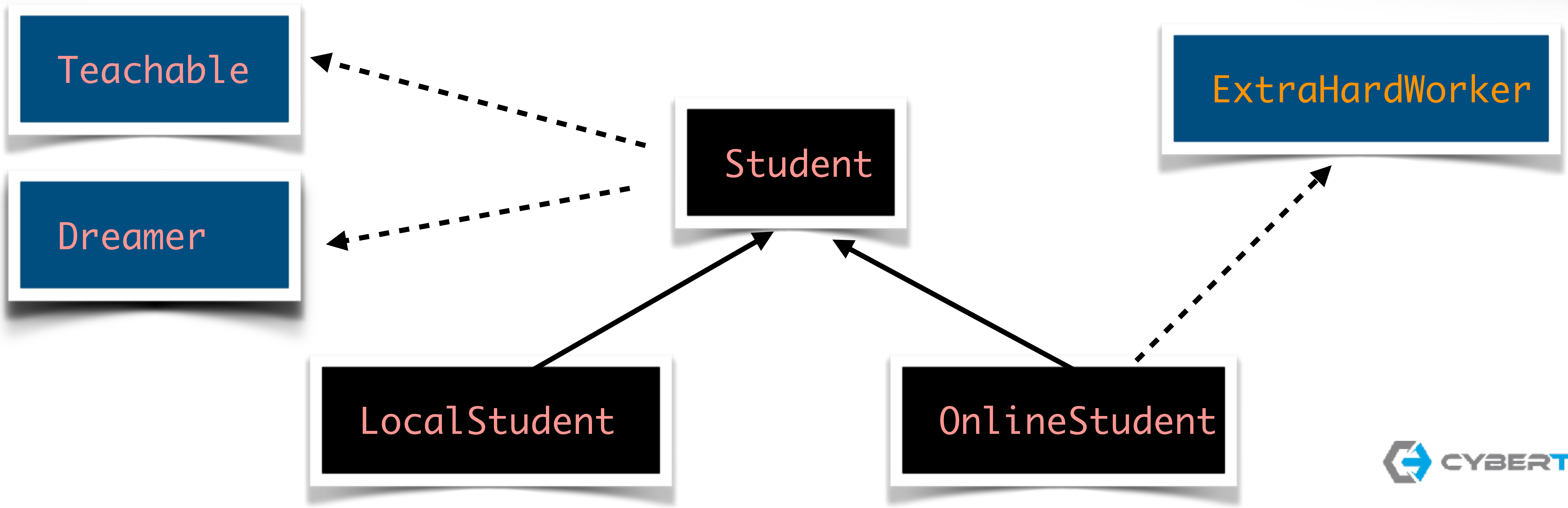
```
public interface ExtraHardWorker { //valid code here }
```

IS-A Relationship

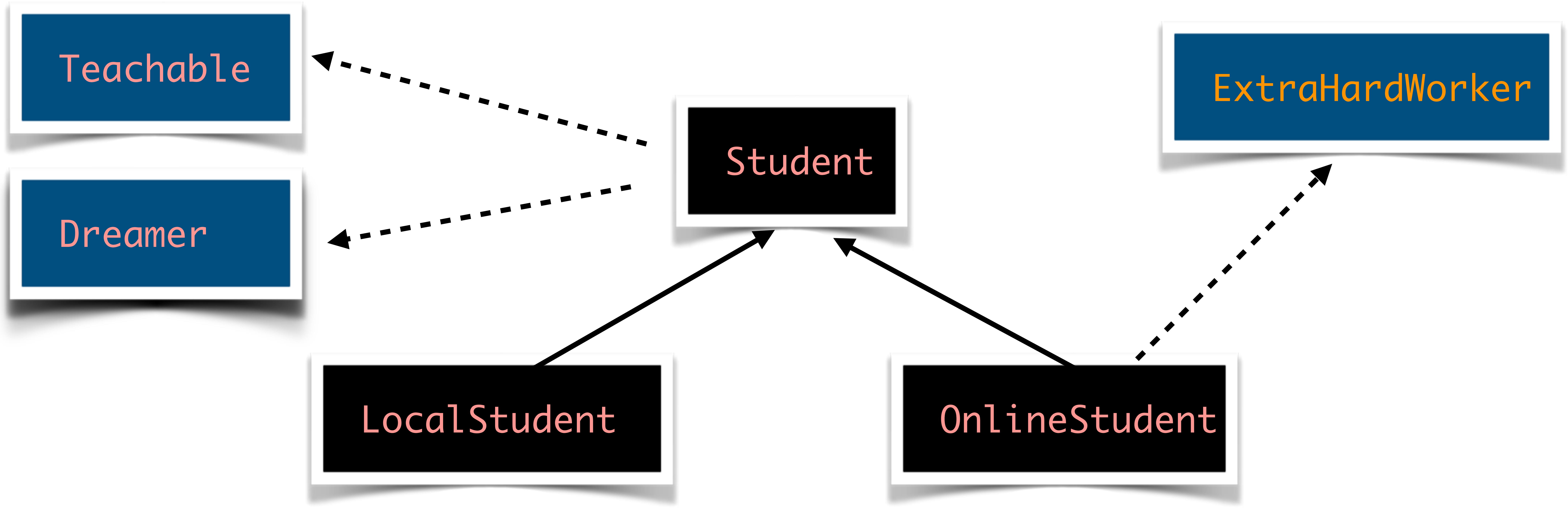
```
public class Student implements Teachable, Dreamer { // valid code here }
```

```
public class LocalStudent extends Student { // valid code }
```

```
public class OnlineStudent extends Student implements ExtraHardWorker{ //valid }
```



IS-A Relationship



IS-A Relationship

- Student is Teachable
- Student is Dreamer
- LocalStudent is Student
- OnlineStudent is Student
- OnlineStudent is ExtraHardWorker
- LocalStudent is Teachable
- LocalStudent is ExtraHardWorker
- OnlineStudent is Dreamer
- Student is OnlineStudent
- …………..

Polymorphism Definition

“Many Form”

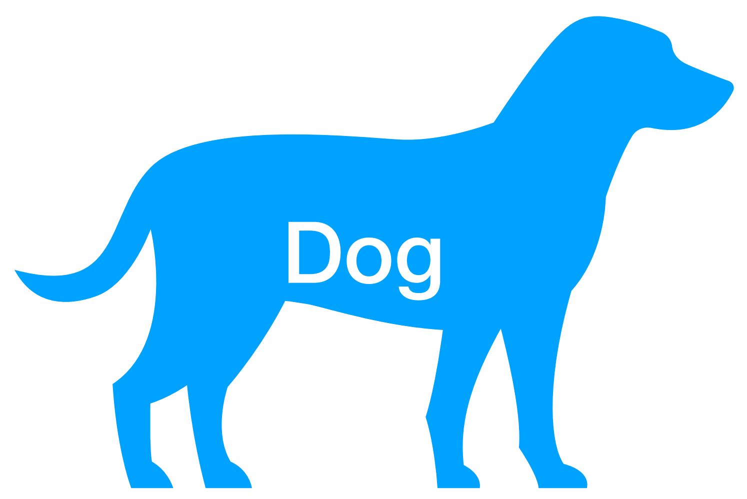
Polymorphism In OOP

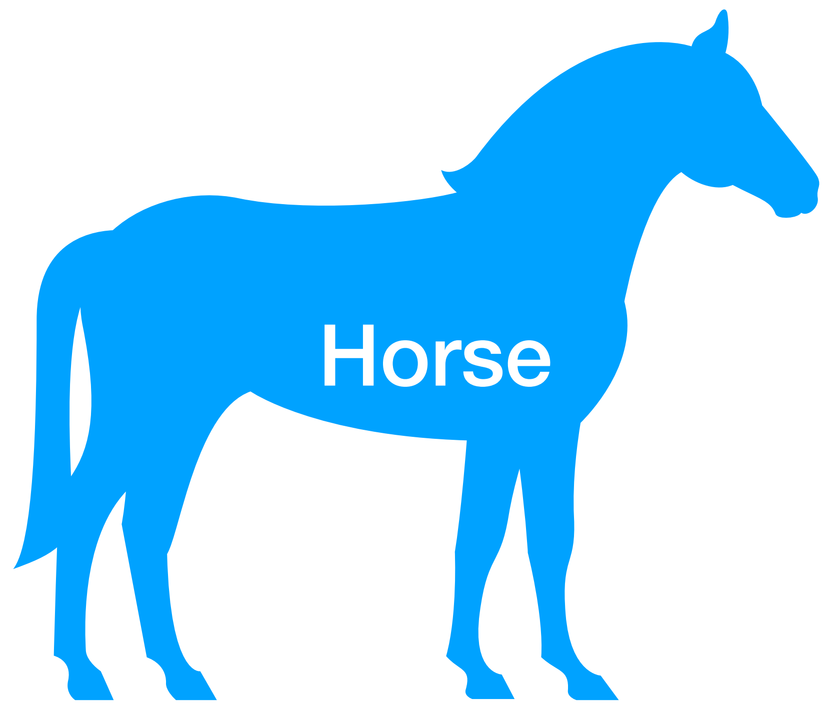
- **Polymorphism** is the ability of an object to take on many forms.

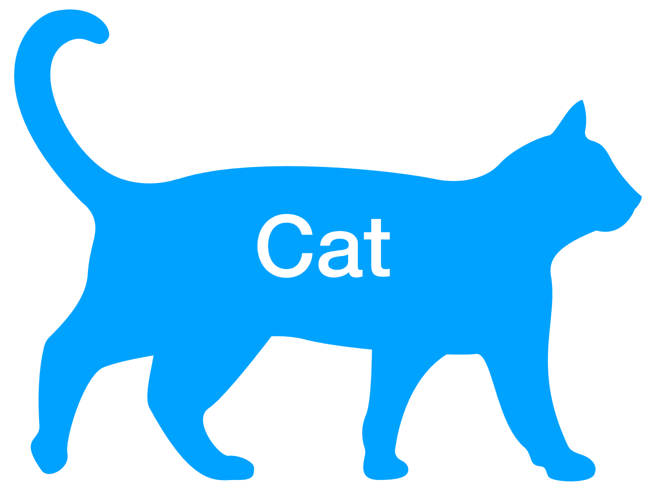
Polymorphism Example

Animal implements Mammal

```
Interface Mammal{
void eat();
}
```







```
Animal a1 = new Dog() ;
```

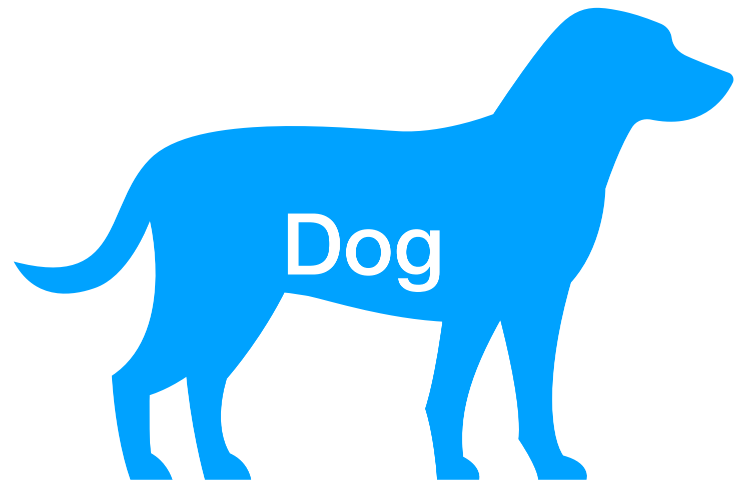
```
Animal a3 = new Cat() ;
```

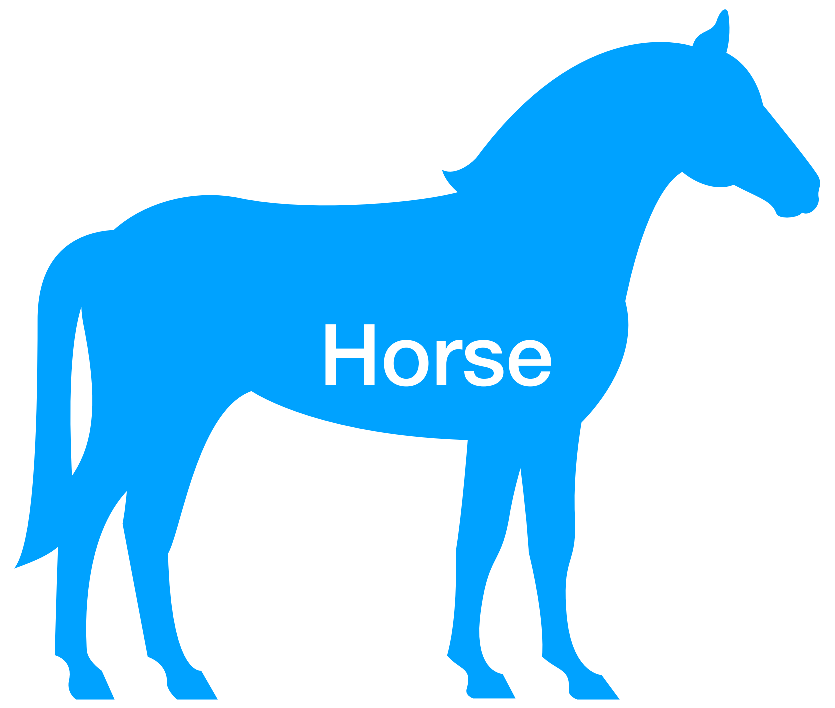
```
Animal a2 = new Horse() ;
```

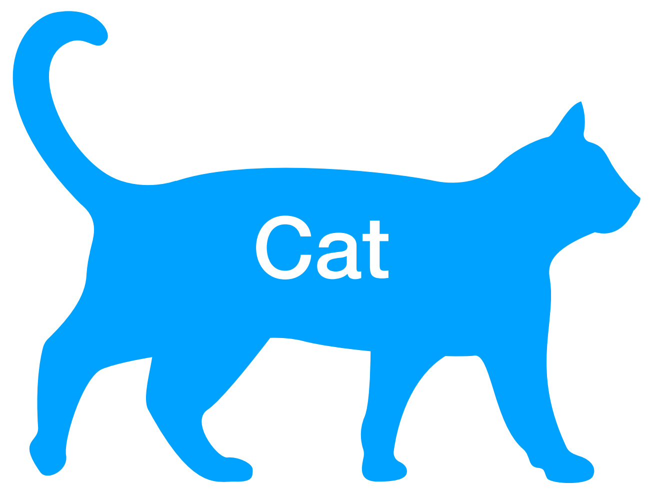
CYBERTEK

Polymorphism Example

Animal **makeNoise() {}**







makeNoise() { //dog noise}

makeNoise() { //Cat noise}

makeNoise() { //horse noise}