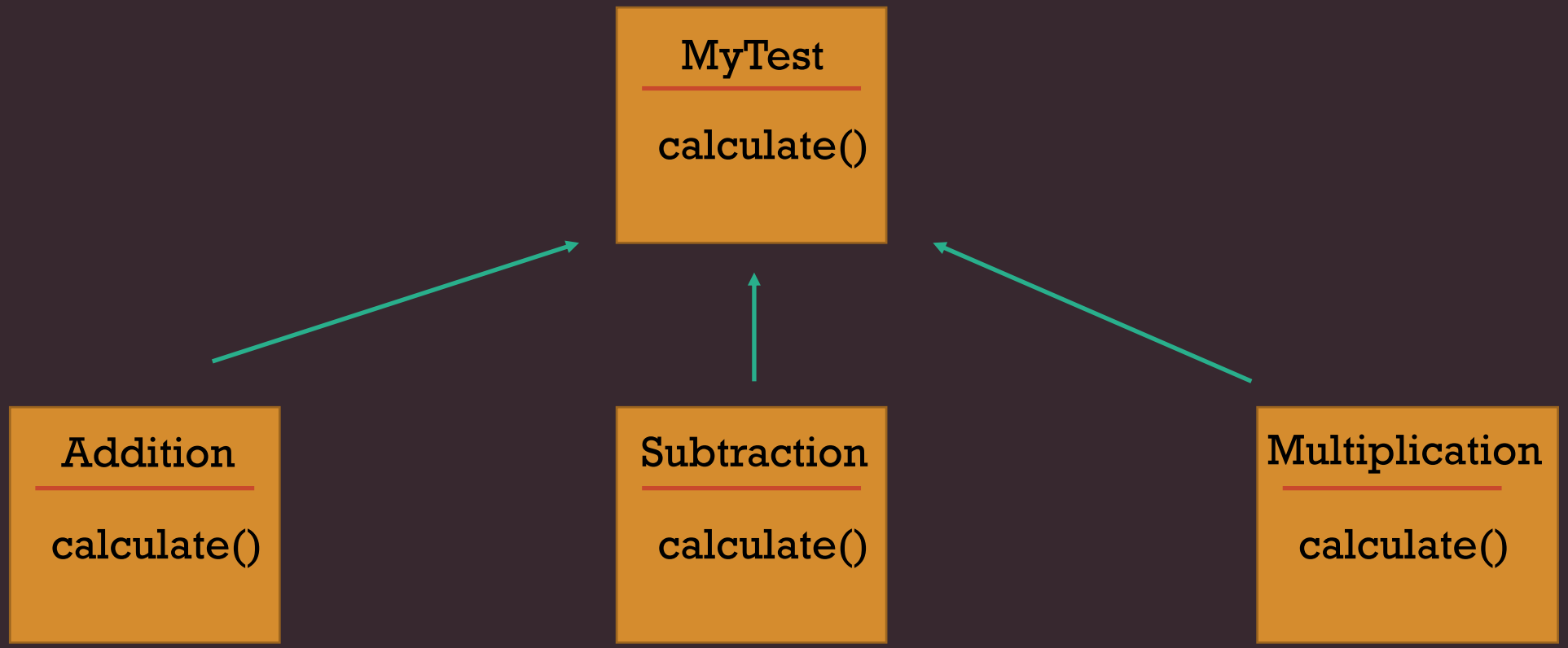
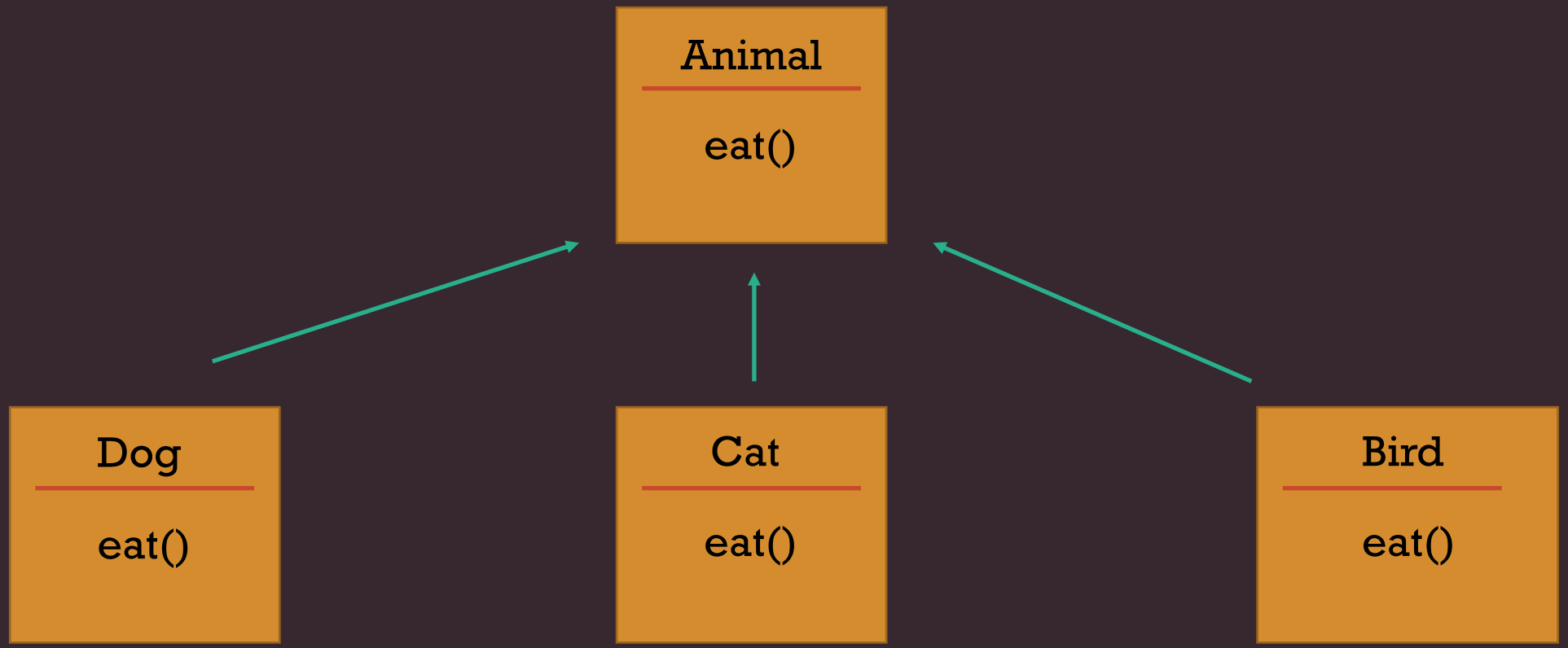
ABSTRACTION


MyTest
calculate()
Addition
calculate()
Subtraction
calculate()
Multiplication
calculate()


Animal
eat()
Dog
eat()
Cat
eat()
Bird
eat()

Abstraction

- Abstraction means we focus on the essential qualities of something rather than one specific example
  - Focus on the essential
  - Ignore the irrelevant
  - Ignore the unimportant

- In Java, abstraction is achieved by abstract classes and interfaces. We can achieve 100% abstraction using interfaces.

Creating Abstract Classes

- abstract keyword is used to create abstract classes
- An abstract class can not be instantiated
- Goal is to provide reusable variables and methods to sub classes

```
public abstract class Student{


}
```

Creating Abstract Methods

- abstract keyword is used to create abstract method
- Abstract method does not have body, only have signature

```
public abstract class Student{

    public abstract void attendClass();

}
```

Creating Concrete Class

- A sub class of abstract class is called concrete class
- A first concrete class must implement all inherited abstract methods

```
public abstract class Student{

    public abstract void attendClass();

}


public class LocalStudent extends Student{

    @Override
    public void attendClass(){

        System.out.println("Attending in person");

    }

}
```

Extending Another Abstract Class

- An abstract class can extend another abstract class. If so it is optional to implement abstract methods from abstract super class.
- A first concrete class must implement all inherited abstract methods.

```
public abstract class Student{

    public abstract void attendClass();

}


public abstract class LocalStudent extends Student{

}
```

Abstract Class Rules Review

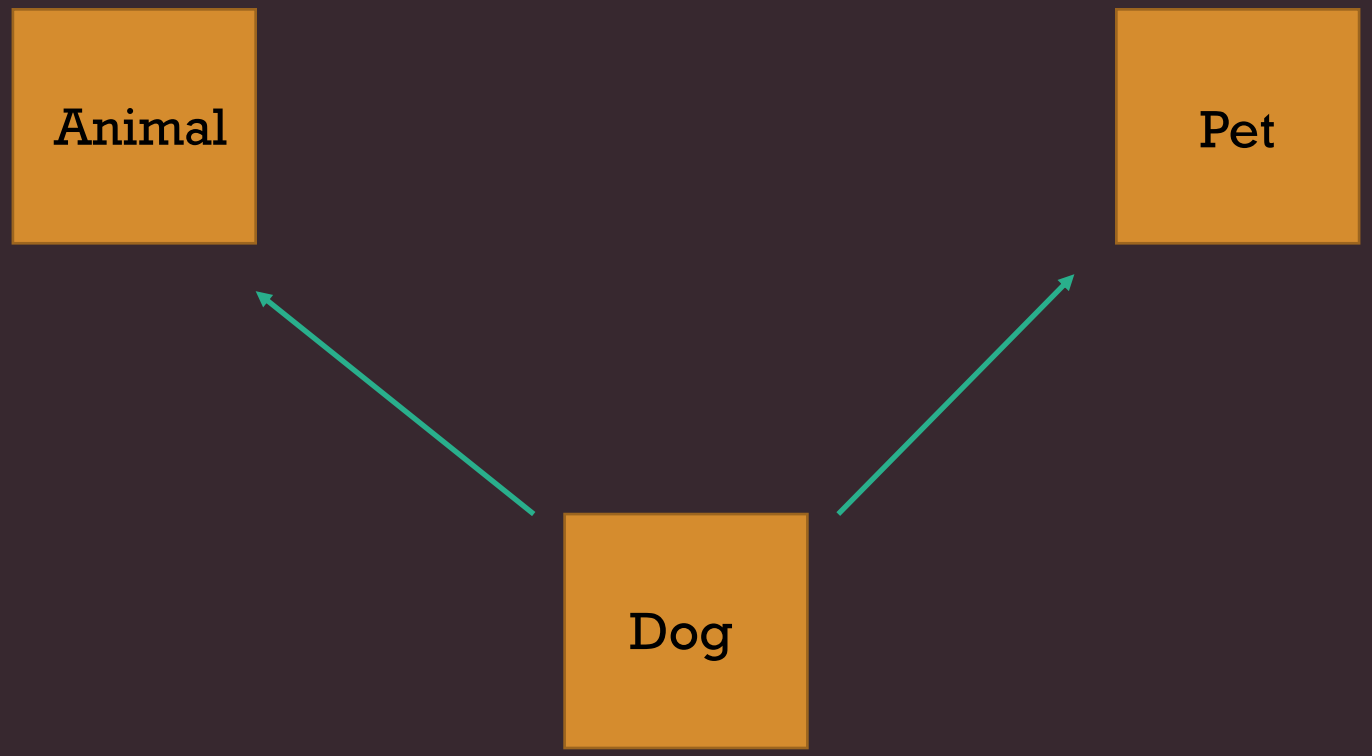
- Abstract classes can not be instantiated directly
- Abstract classes may be defined with any number, including zero, of abstract and non-abstract methods
- Abstract classes may not be marked as private or final
- An abstract class that extends another abstract class inherits all of its abstract methods as its own abstract methods
- The first concrete class that extends an abstract class must provide an implementation for all of the inherited abstract methods

Abstract Method Rules Review

- Abstract methods may only be defined in abstract classes
- Abstract methods may not be declared private or final
- Abstract methods must not provide a method body/implementation in the abstract class for which it is declared
- Implementing an abstract method in a subclass follows the same rules for overriding a method

Interface

Is it possible to extend 2 classes in Java?



Interface

- Contract between a class and outside world
- Provide set of abstract methods
- Most interfaces have a group of related empty methods
- The class provides the behaviors included in the interface

Creating an Interface

An interface looks similar to a class, except the keyword interface is used instead of the keyword class

```
public interface Teachable{


}
```

Interface

- Methods that are specified in an interface have no bodies.

```
public interface Teachable{

    public abstract void canLean();
    public abstract void doHomework();

}
```

Interface

- When you want a class to implement an interface, you use the implements keyword.

```
public interface Displayable{

    public abstract void display();

}


public class Person implements Displayable{

    public void display(){

        //code
    }
}
```

Implementing Multiple Interfaces

- Class can extend only one superclass, but java allows a class to implement multiple interfaces.
- When a class implements multiple interfaces, it must provide the methods specified by all of them

```
public class MyClass implements Interfacse1,Interfacse2,Interfacse3{}
```

Fields in Interface

- An interface can contain field declarations, but all fields in an interface are treated as final and static.

```
public interface Interface{

    public static final int field1=5;

}
```

Implementing Interface

- A class can extends another class and implements interface(s) same time.
- If a class both extend a class and implement an interface, extends should come first then implements keyword.

```
public class Student extends Person implements Teachable,Dreamer{


}
```

Default and Static Methods

- Beginning in Java 8, interfaces can have default and static methods.
- Default method is an interface method that has a body.

```
public interface Teachable{

    public default void readTopics(){
        //code
    }

    public static int doResearch(){
        //code
    }
}
```

- Interfaces can have:
  - Constant variables
  - Abstract methods
  - Default Methods
  - Static Methods
- Interfaces can not have:
  - Constructor
  - Blocks
  - Instance variables or methods