Philip Pesic

Week 10

March 26 2023

Week 10 Lecture Notes

Inheritance

Similar to composition, inheritance is another form of class relationship. Composition, however, is a HAS A relationship, where a class has or is composed of many elements, whereas inheritance is an IS A relationship, where a class IS another class. With inheritance, subclasses or specific classes inherit the properties of base/general/super classes.

Ex:

A kid IS A human

A basketball IS A ball

Uses of inheritance

The main reason to use inheritance is for code reuse. By having a subclass or many subclasses inherit the methods of a base class, you dont need to write the same methods for each subclass. Instead, you can write common methods once in a super class, and then inherit them in subclasses.

Ex:

```
Public class ball {
//List 10 methods for a ball
}
Public class basketball extends ball {
//Instead of relisting methods, methods from ball are automatically inherited and can be used
}
```

Philip Pesic

Week 10

March 26 2023

Week 10 Lecture Notes

Using Inheritance

There are 3 steps to using inheritance:

1: Define super class

2: Define subclass

3: Use subclass methods

Ex:

```
Public class car { //Define super/general class
        Public void wheels() {
                System.out.printn(“I have wheels”);
        }
}




Public class tesla extends car { //Define subclass that inherits from super class
        Public void engine() {
                System.out.println(“I have an electric engine”);
        }
}
```

Philip Pesic

Week 10

March 26 2023

Week 10 Lecture Notes

```
Class main {
        Public static void main(String[] args) {
        tesla t = new tesla();
        t.wheels(); //Gets method from super class car
        t.engine(); //Gets subclass specific method
        }
}
```