Inheritance:

* Rectangle is a sub class of shape
* Rectangle has all the methods of the Shape class, since it is a sub class
* Super() allows us to use the Shape class since Shape is the super class of rectangle
* Shape -> Rectangle -> Sqaure

Abstract class:

* The purpose is to be a super class, can make Animal variables
* Can’t make objects of Animal
* Abstract class can have all those of a regular class except has an abstract method
* Abstract method does not have a code but must include it within the new class made
* In LoudDog, when we return super.getSound(), we are overwriting the getSound() in the super class with the method we have in LoudDog
* If we do not include a constructor in any additional class to super class, automatically calls the default constructor of the super class

// Inserts public LoudDog(){

super(); // If super class does not have a default constructor, then must write the line

}

* Object is the highest in the hierarchy meaning, for example, when declaring a new shape, it calls the constructor of shape, shape calls the constructor of Object

|  |  |  |  |
| --- | --- | --- | --- |
|  | Concrete Class | Abstract Class | Interface |
| fields | yes | yes |  |
| constructors | yes | yes |  |
| Concrete methods | yes | yes |  |
| Abstract methods | no | yes | yes |

Interface:

* Has one super class
* Final means the variable cannot be changed
* Static means that it is a global variable that changes
* Both static and final are constants