An example of using “%”: we need to load a certain number of boxes to a truck, and the box each has a fixed limit number of units (assume we use the same type of box so the limit is the same). The truck, too, has a limit of units to carry. In this case, we can use “%” to find out the number of units of the box with least units.

1. Difference: overloaded methods can be static(use static binding) but overridden methods can’t be static(use dynamic binding); overloaded methods can be private and final but overridden methods can’t be; return type of overloaded methods can be different because more than one method shares the same name with different signatures but return type of overridden methods should be more specific.
2. The type int has a smaller range than double, so when we try to execute int i = 0; double d = i; it is type widening and would work. However, int i = d; is type narrowing and we can’t be sure that d is in the range of i so this would give an error.

Fix it by type casting: int i = (int) d;

Type casting would work because we cast the type double into int by truncating d, making sure the truncated d lies in the range of int.

1. equals() compares values of objects while == compare whether the tow objects are the same object(which have the same pointer).
2. We keep certain fields private to protect data and make code easier to maintain; private methods use static binding so it’s faster and could optimize compile time; private variables can also provide getters and setters.

Vegetable

LeafVegetable RootVegetable

Arugula Lettuce Spinach Carrot Beet TuberVegetable

Yam Potato

1. Class represents a strict “is - a” relationship while interface allows us to add additional features of a class. When we want to add common features to different classes, we can implement an interface to do so.
2. this keyword is the name of a reference that refers to an object itself. Within a constructor or an instance method, this is used to refer to the current object. It can be used to call other methods in the same class.
3. The two types of errors are IOException and Runtime Exception. IOException is a checked exception and checked at compile time, the compiler forces us to check and deal with it (throw Exception or write try/catch blocks), then the program will compile. Runtime Exception falls in the category of Unchecked Exception, which aren’t be checked and dealt with at compile time.
4. GeometricObject is superclass and Circle is subclass, so Circle inherits GeometricObject’s methods and attributes. In the superclass, color has been set to “white” in the no-argument constructor. When we execute Circle c = new Circle();the radius of the circle has been set to 1.0 inside the no-argument constructor of Circle, while this new circle inherits color and filled from GeometricObject with superclass’s no-argument constructor, the color is set to white.
5. By executing Person p2 = *changePerson*(p1); it uses the changePerson method, which tells us that Person p1 and Person p2 are the same object (Person) and the age is 25. John is Person p1 and Person p2, so p1.age and p2.age are 25.