

# Other

---

## What is a class?

A class is a template for creating objects. It is a collection of variables(data) and methods(behavior). It is a user defined data type.

```
public class Dog {  
    private String name;  
    private int age;  
    private float weight;  
  
    public Dog(String name, int age, float weight){  
        this.name = name;  
        this.age = age;  
        this.weight = weight;  
    }  
  
    public void eat(){  
        System.out.println(this.name + " is eating");  
    }  
  
    public String getName(){  
        return this.name;  
    }  
  
    public void setName(String name){  
        this.name = name;  
    }  
}
```

---

## What is an object?

An object is an instance of a class.

```
public class Main {  
    public static void main(String[] args) {  
        Dog tom = new Dog("tom" , 10 , 40.0f);  
        tom.eat();  
        System.out.println(tom.getName());  
        tom.setName("bob");  
        System.out.println(tom.getName());  
    }  
}
```

```
}
```

---

## What is a method?

A method is a action or behavior that an object can perform. It is defined inside a class and by default belongs to the object, using the static key word it makes it belong to the class. Block of code that does something

---

## Commonly used classes that exist as a part of the standard library

- String - A sequence of characters (piece of text)
  - Object - Super class of all classes
  - List - An interface that represents of lists
  - ArrayList - A class that implements the List interface
  - System - A class that contains methods that are used to interact with the system
  - Math - A class that contains methods that are used to perform mathematical operations
  - Scanner - A class that is used to read input from the user
  - File - A class that is used to represent a file
  - Date - A class that is used to represent a date
- 

## Decoupling vs Coupling

Decoupling means reducing the interdependence between different parts of the code. When two components are tightly coupled, they are highly dependent on each other, and changes made to one component can cause unexpected behavior or even break the other component. On the other hand, when components are decoupled, they are less dependent on each other, and changes made to one component do not affect the behavior of the other component as much. Making the dependencies easier replaced or modified without affecting the behavior of other objects in the systems.