Bank Soal OFSE Programming

# Java Class and Methods

## What is meant by garbage collection?

Garbage collection is the process of deconstructing unused objects or objects without reference to save memory. In Java, this process is automatically done by having the virtual machine of Java constantly checking for objects with no reference. Objects with no reference, will automatically be deconstructed.

James:

Suatu mekanisme yang dimiliki oleh Java vm untuk menghapus simpanan-simpanan memory dari object yang tidak dipakai lagi.

## What are methods and how are they defined?

A method in Java is a block of code that, when called, performs specific actions mentioned in it. For instance, if you have written instructions to draw a circle in the method, it will do that task. You can insert values or parameters into methods, and they will only be executed when called.

James:

Blok kode yang bisa dijalankan apa bila dipanggil.

Bagaimana cara menulisnya?

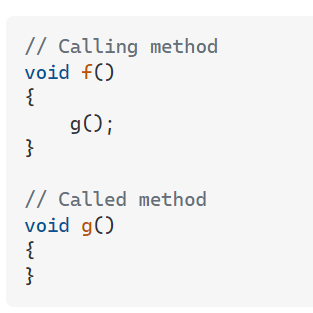
Access modifier, return type (void=gaada return type, int|String= return type), method signature, parameter, list of commands | body

## What is calling method? Give an example!

The calling method is the method that is being called. The callee is the method that calls the called method.

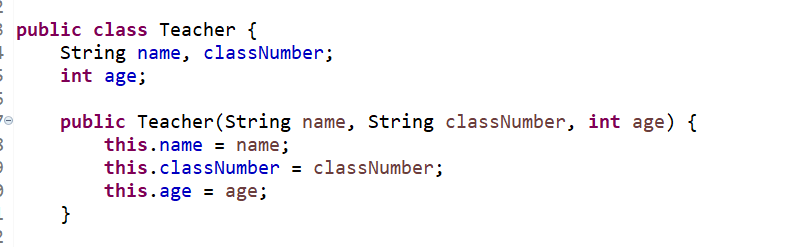
James:

Kita memanggil untuk menjalankan method.



## What is a constructor? Give an example!

It is a method that is used to initialize an object. The constructor is called when an object of a class is created.

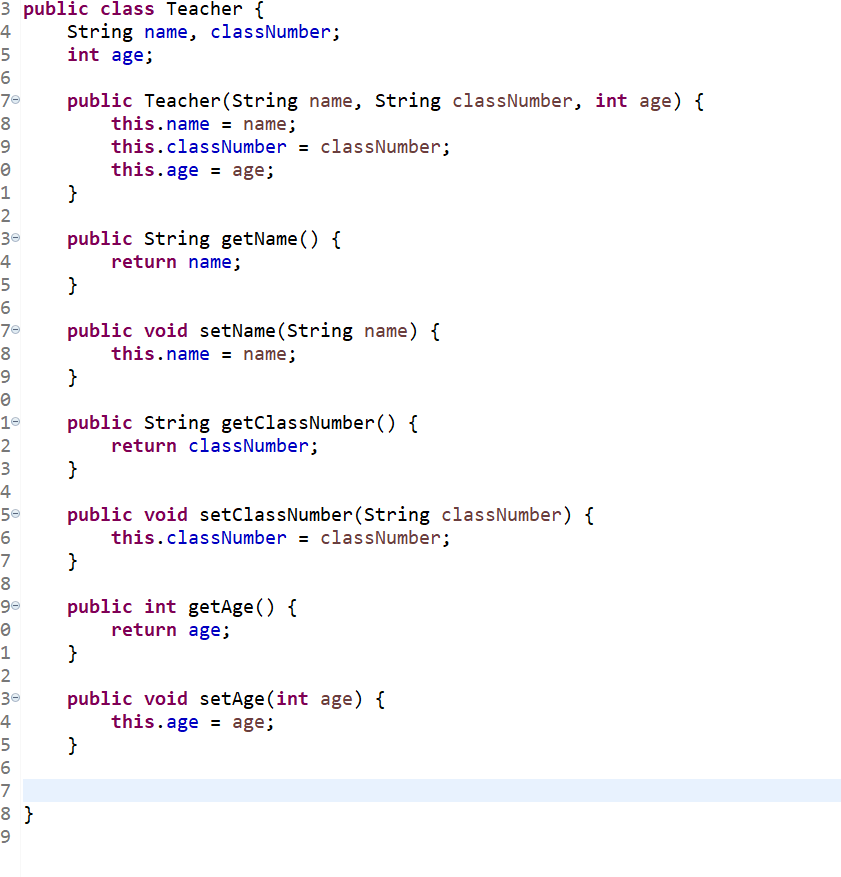


James:

Method untuk meng-initialize suatu object, nama method harus sama dengan nama class.

## Suppose a given instance variable has been declared private. Can this instance variable manipulated by methods outside its class? Explain your answer. (a.yes, b.no)

No, private methods cannot be directly be manipulated from outside of its class. However, it can be manipulated from a method from inside its class. For example, using a “setter” and “getter” methods.



## Which of the following statements can be used to describe a public method? (It is accessible to all other classes in the hierarchy, It is accessible only to subclasses of its parent class, It represents the public interface of its class the only way to gain access to this method is by calling one of the public class methods)

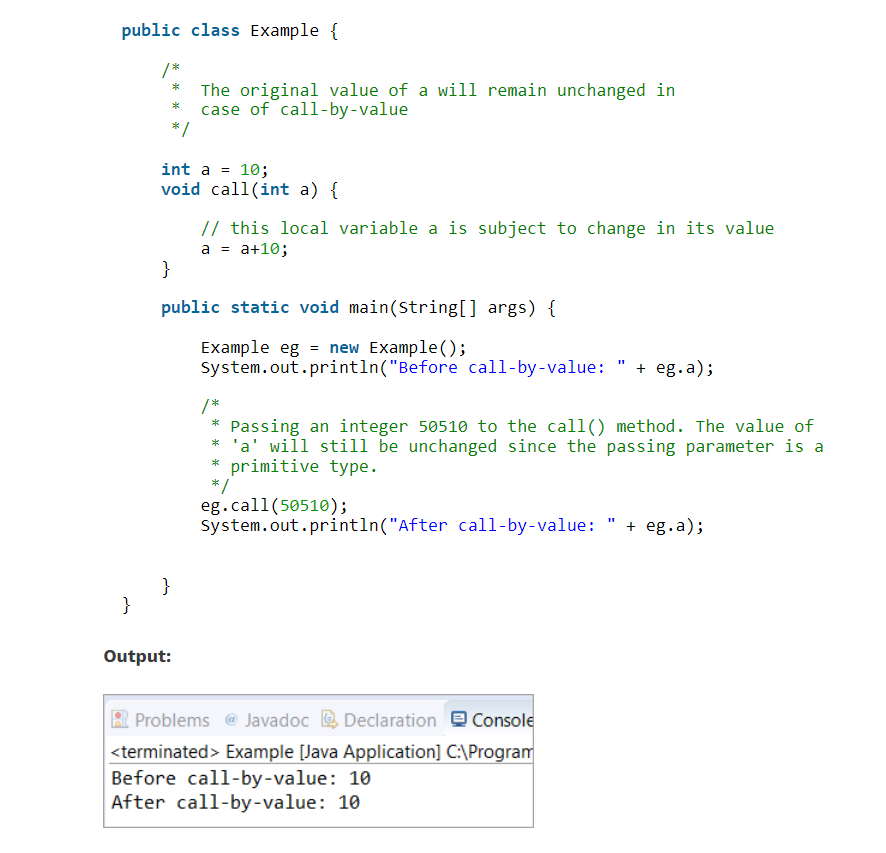
It is accessible to all other classes in the hierarchy.

## What is meant by "Passing by value" and " Passing by reference"?

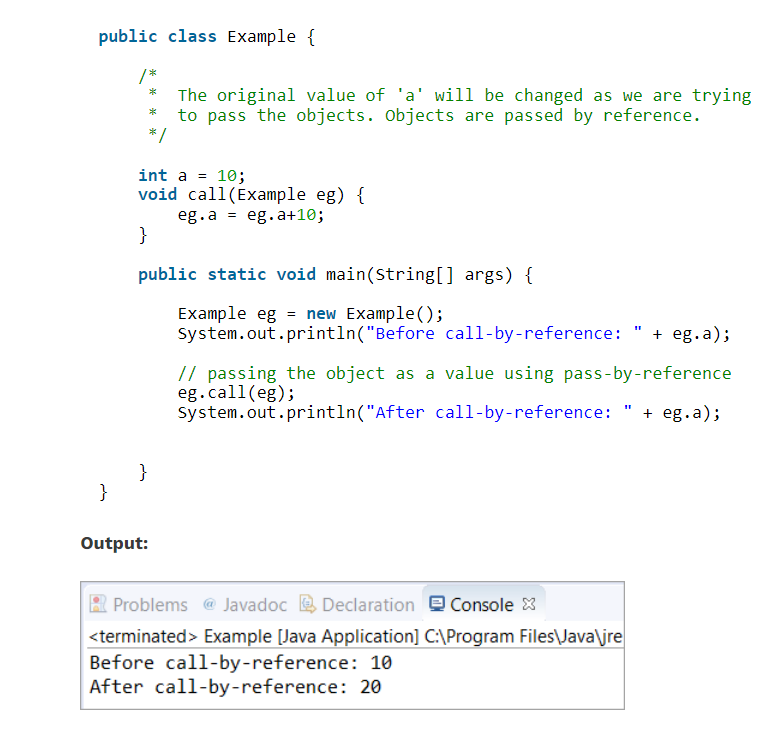
Pass by Value: It is a process in which the function parameter values are copied to another variable and instead this object copied is passed. This is known as call by Value.

Pass by Reference: It is a process in which the actual copy of reference is passed to the function.

Pass by value example:



Pass by reference example:



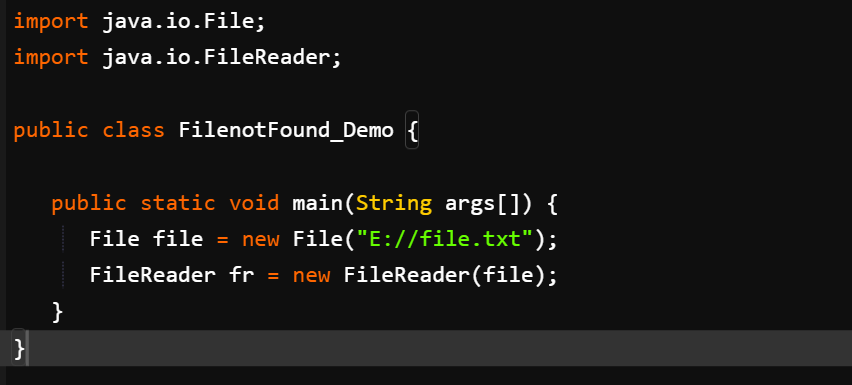
## What is an “unchecked runtime exception”?

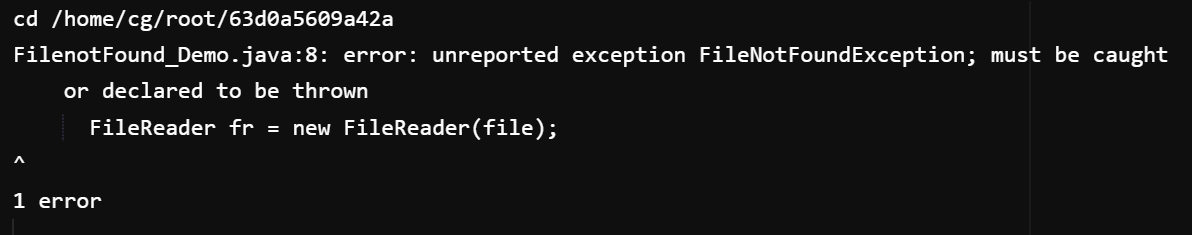
James:

Unchecked runtime exception adalah error yang tidak terdeteksi pada saat compile

A checked exception is an exception that occurs at the compile time, these are also called as compile time exceptions. These exceptions cannot simply be ignored at the time of compilation, the programmer should take care of (handle) these exceptions.

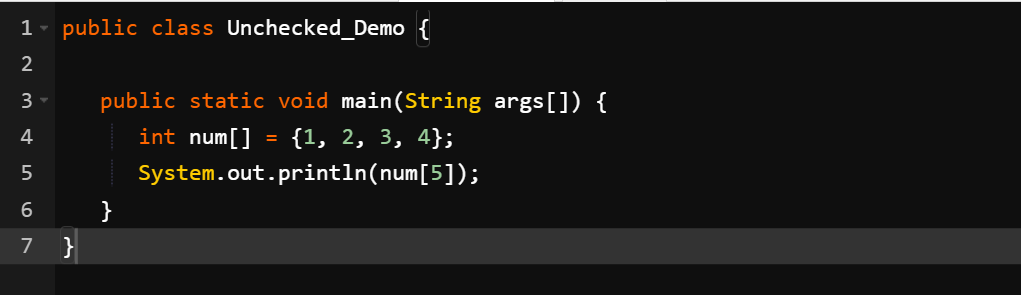
Checked exception example:

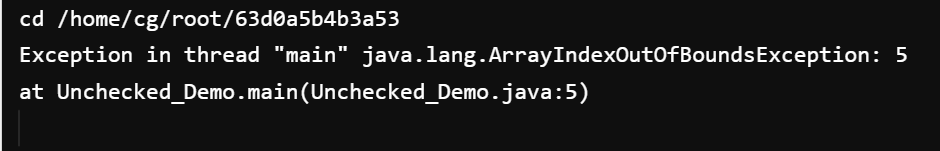




An unchecked exception is an exception that occurs at the time of execution. These are also called as Runtime Exceptions. These include programming bugs, such as logic errors or improper use of an API. Runtime exceptions are ignored at the time of compilation.

Unchecked runtime exception example:

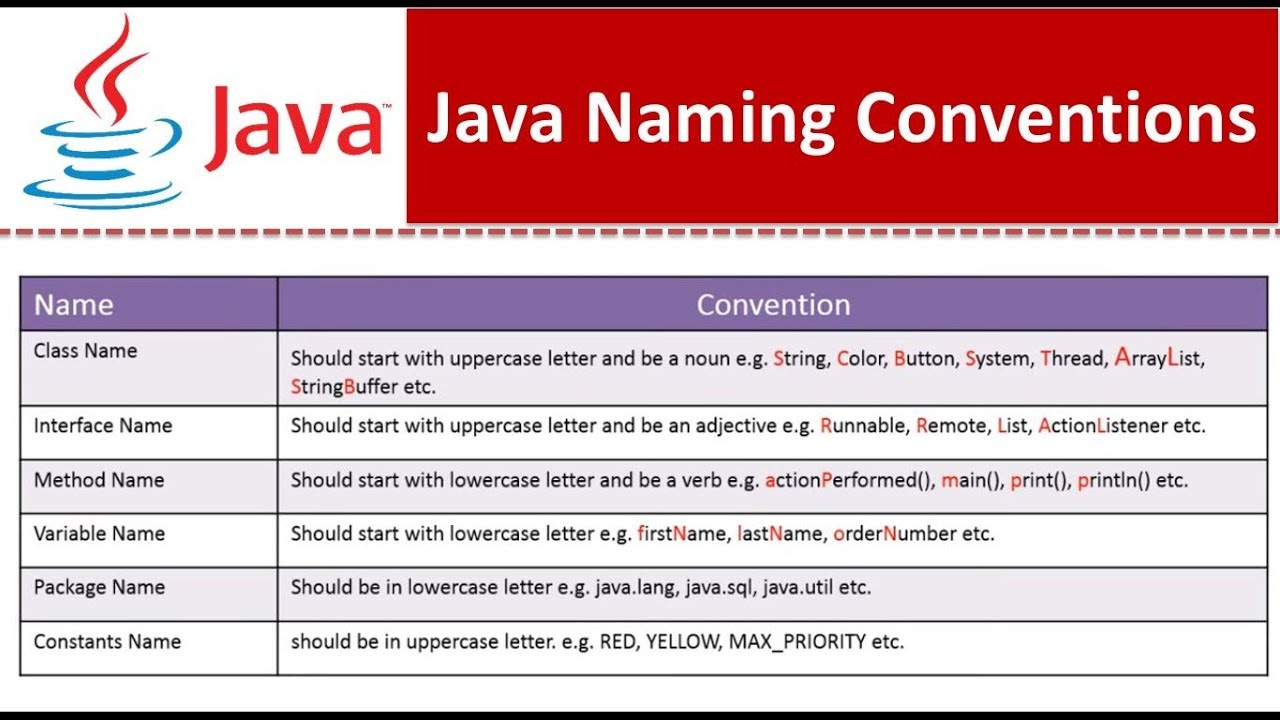




## What are identifiers and what is naming convention?

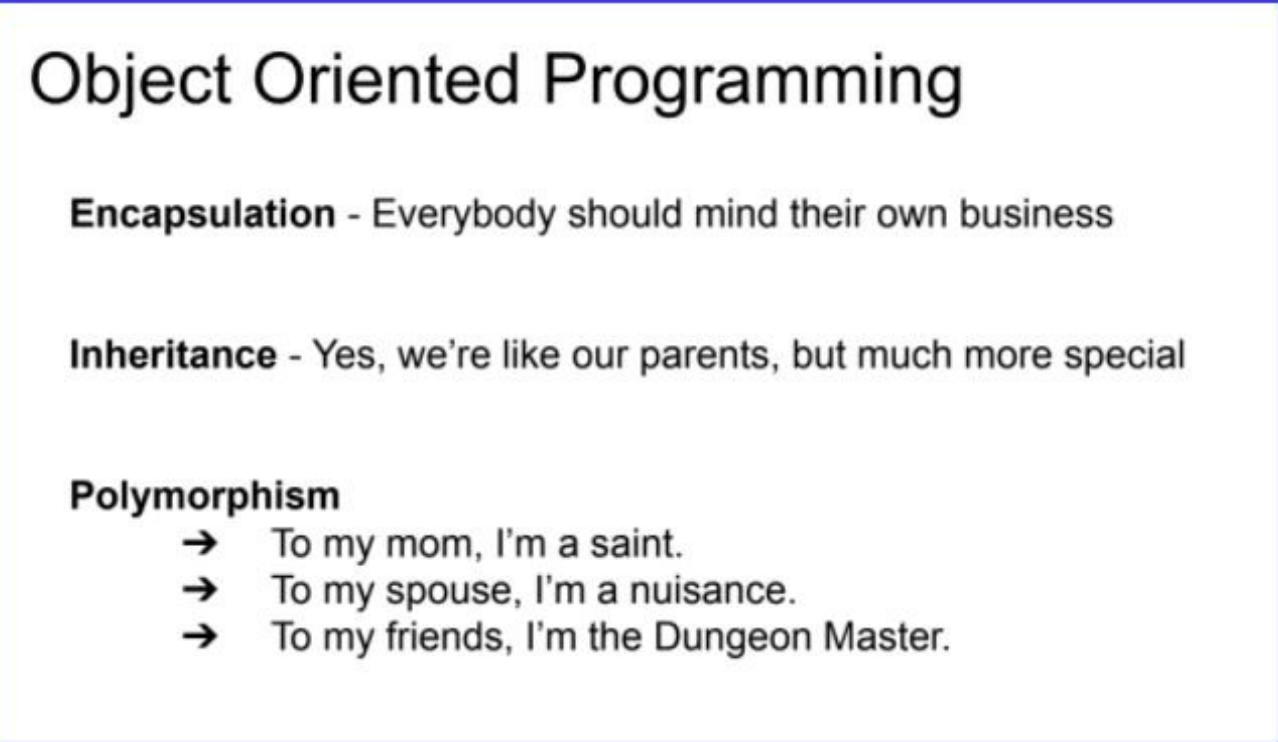
Identifiers in Java are a sequence of characters to identify something in a program. They are names given to a class, variable, package, method, or interface and allow the programmer to refer to the specific item from any place in the program.

Java naming convention is a rule to follow as you decide what to name your identifiers such as class, package, variable, constant, method, etc. But it is not forced to follow. So, it is known as convention not rule. These conventions are suggested by several Java communities such as Sun Microsystems and Netscape.



## What are the three OOPs principles and define them?

There are three major pillars on which object-oriented programming relies: **encapsulation**, **inheritance**, and **polymorphism**.



**Encapsulation**: This is the idea of wrapping everything up about a particular thing, whether a Checking Account or Armadillo, into a defined object with features and behaviors.

Once we do, we can ask the object itself to do what it is supposed to do, whether that is DepositMoney or DefendYourself. But nobody outside the object needs to worry about how it does its jobs. We just tell it to do it and go about our day. If everybody, and by this, I mean every object, simply minds its own business and stays out of the business of other objects, all is good with the world.

**Inheritance**: This is the idea that we do not have to define absolutely everything about an object repeatedly if it shares features and behaviors with other objects.

We can define a class for Accounts and then let our Checking Account or Savings Account inherit all the stuff in common. Likewise, we can define a class for Animals, and let our Armadillo inherit features like NumberOfLegs and Weight as well as behaviors such as Breathe and Sleep. We call these overarching classes parent classes, and the ones that inherit from them, child classes. We can then inherit from the child classes and so on. But our Checking Account is more specialized than our Accounts because we can WriteACheck, which we can’t do with a Savings Account. Our Armadillo can RollIntoABall, but other animals such as a Giraffe don’t have that behavior. Since we go from more general to more specialized, I like to say that a child is like its parents, but much more special.

**Polymorphism**: This fancy name just means that we can treat the same object as different things depending on how we need it at different times, and we can treat groups of different objects that share an ancestor or trait as if they were that ancestor or trait.

So, we could have a set of different Checking, Savings, and Credit Accounts and ask each to GetBalance so we can figure out how much we have to spend on vacation this year. Or we could ask a queue of animals to MoveQuickly, and not care how the Porpoise or Eagle or Armadillo would handle that shared behavior. I like to think that we are different things to different people, so even if not every Dungeon Master has a spouse to think him or her a nuisance, we can ask any of them to organize a game for Saturday night.

## Explain the difference between an object and an object reference!

## Explain the difference between an object and a class!