

Lecture 4: Constructor and method, Variables

Difference between constructor and method in java:

Constructor	Method
Constructor is a special method of a class but can't be invoked directly like method call. Constructors are invoked implicitly	Methods are invoked explicitly.
It has the same name as its class name.	Can have same name as its class name, but the existence of return type makes it a method (unfortunately looks like constructor).
It is not a member of a class as it can neither be inherited nor invoked using dot (.) operator.	Dot (.) operator is used to invoke non-static methods via objects and static methods via class name.
It has no explicit return type.	It has explicit return type, if there is nothing to return, the return type must be void.
It is used to initialize an object that is being created with the new operator.	Method is used to exhibits functionality of an object.
In case constructor is not present, a default constructor is provided by java compiler.	In the case of a method, no default method is provided.
A constructor can never be abstract or static.	A method is of two types defined (implemented) or undefined (abstract).The method implementation can be further categorized as static on non -static. An abstract method can't be static or final.
Can be specified as public, none (default), protected or private.	Access specifiers public, none (default), protected or private are applicable.
Can't be final, native or synchronized.	Can be final, native, static or synchronized.
E.g. class T{T(){} }	E.g. class T{ void T(){} }

Variables in Java:

Local variable:

Local variables are those which are declared **within any block of code like methods, constructor or initialization block**. A block defines a scope. We cannot use access specifier in local variable.

```
E.g. class T{  
    psvm(...){  
        int i;  
    }  
}
```

Member Variable:

Member variables in a class called **fields**.

There are two kinds of member variable: instance and static.

Instance variable:

A variable declared inside the class but outside the body of the method, is called instance variable. It is not declared as static. It has **one copy per instance of a class**.

```
E.g. class T{  
        int i;  
}
```

Static variable:

Instance variables declared as **static (global like)** at the class level are, essentially, global variables.

It has **one copy per class** so it saves memory.

```
E.g. class T{  
        static int i;  
}
```