# Constructor

A **constructor** is used in the creation of an object, that's an instance of a class.

It is a special type of code block that has a specific name and parameters, much like a method.

It has the same name as the class itself, and it doesn't return any values.

You never include a return type from a constructor, not even void.

You can, and should, specify an appropriate access modifier, to control who should be able to create new instances of the class.

```
public class Account {  // This is the class declaration

    public Account() {  // This is the constructor declaration
        // Constructor code is code to be executed as the object is created.
    }
}
```

# The default constructor

If a class contains no constructor declarations, then a default constructor is implicitly declared.

This constructor has no parameters, and is often called the no-args (no arguments) constructor.

If a class contains any other constructor declarations, then a default constructor is NOT implicitly declared.

# Constructor overloading

Constructor overloading is declaring multiple constructors, with different formal parameters.

The number of parameters can be different between constructors.

Or if the number of parameters is the same between two constructors, their types or order of the types must differ.