**MATRIC**: **AUL/SCI/18/00256**

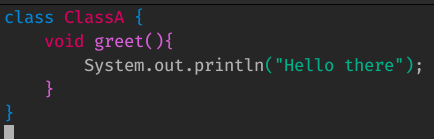
**NAME**: **TEKENA SOLOMON**

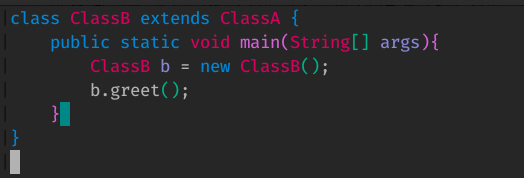
1. Explain (with examples) the following:
2. Inheritance
3. Class
4. Object
5. The \*static\* keyword
6. The \*public\* modifier
7. An abstract class
8. Method overloading
9. Method overriding
10. Constructor
11. A final method.

**SOLUTION**

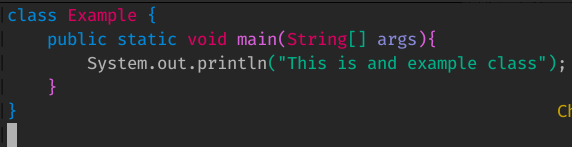
1. Inheritance: Inheritance is a method where one object(class) acquires all the properties and methods of a parent object(class).

**ClassA.java**

 **ClassB.java**

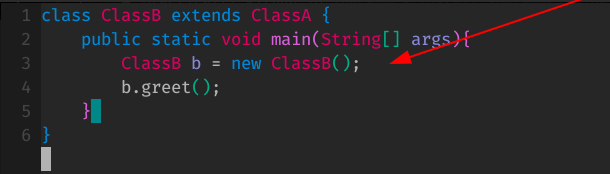


1. Class: A class is a piece of code, *object* constructor which may contain attributes and methods used to create instances of an object.

 **Example.java**

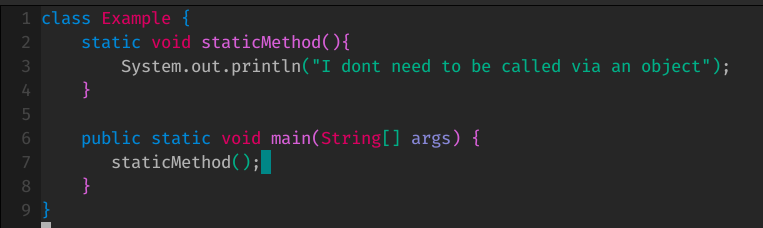
1. Object: An object is an instance of a class, having access to all the methods and properties of that class.

**ClassB.java**

line 3 shows an object *b* as and instance of the class *ClassB*

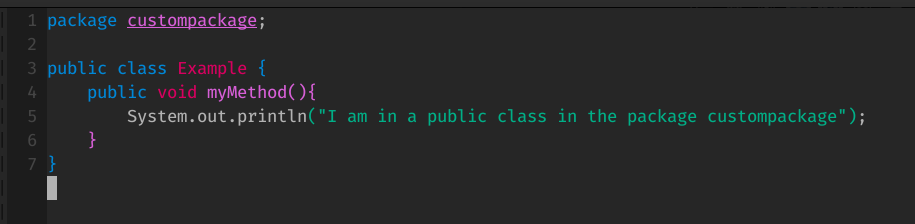
1. The \*static\* keyword: The keyword static is used to create methods that belong to the class rather than an instance of the class. Static methods will not be in an instance of the class.

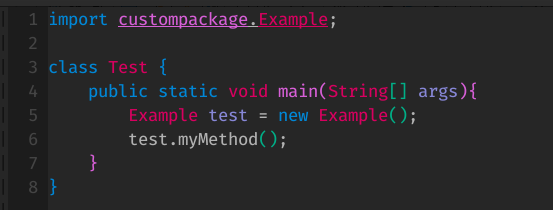
**Example.java**



1. The \*public\* modifier: This is used to make classes or methods visible to other classes or methods respectively. Other classes can modify public fields.

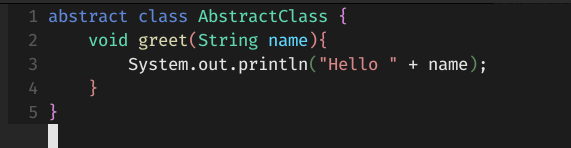
**Example.java**

The class and method above can both be used by an external class.

 **Test.java**

1. Abstract class: An abstract class is a class declared abstract using the *abstract* keyword. These classes cannot be instantiated, meaning objects of that class cannot be created.

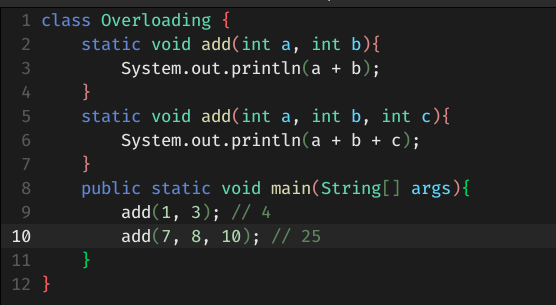
**AbstractClass.java**



The class above cannot be used to create and object.

1. Method overloading: This is the ability to create multiple methods with the same name but different parameters (implementations).

**Overloading.java**



1. Method overriding: This is when a subclass has the same method in the parent class with a specific implementation of the method.

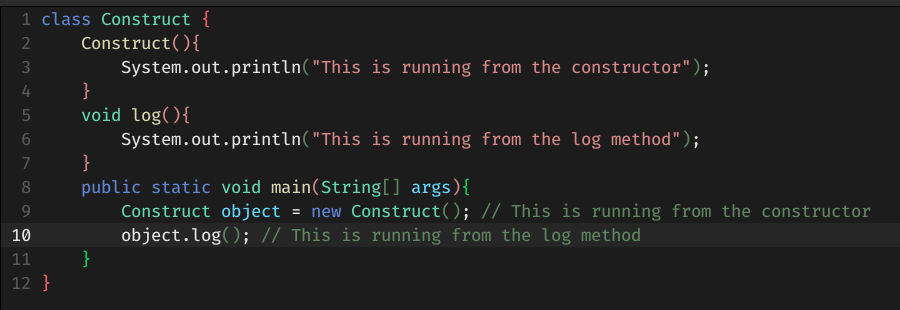
 **TestForFinal.java**

**Ex.java**



1. Constructor: A constructor is a method that has the same name as the class name with no return type and is executed when an object of the class is created.

**Construct.java**



1. Final method: A method declared final cannot be overridden in a subclass (child class). Meanwhile a class declared final cannot be inherited.

**TestForFinal.java**

****