THE TEMPLATE PATTERN

MORE AND MORE HIGH-LEVEL STUFF
IS DONE USING

FRAMEWORKS

FRAMEWORKS ARE COMPLICATED GROUPS
OF CLASSES THAT DO A BUNCH OF STUFF -

AS A PROGRAMMER YOUR JOB
IS TO PLUG IN LITTLE BITS OF CODE
THAT CUSTOMIZE WHAT THE
FRAMEWORK DOES

FRAMEWORKS WIDELY USE SOMETHING
KNOWN AS THE

DEPENDENCY INVERSION PRINCIPLE

In object-oriented programming, the **dependency inversion principle** refers to a specific form of decoupling software modules. When following this principle, the conventional dependency relationships established from high-level, policy-setting modules to low-level, dependency modules are inverted (i.e. reversed), thus rendering high-level modules independent of the low-level module implementation details. The principle states:[1]

> *A. High-level modules should not depend on low-level modules. Both should depend on abstractions.*
>
> *B. Abstractions should not depend on details. Details should depend on abstractions.*

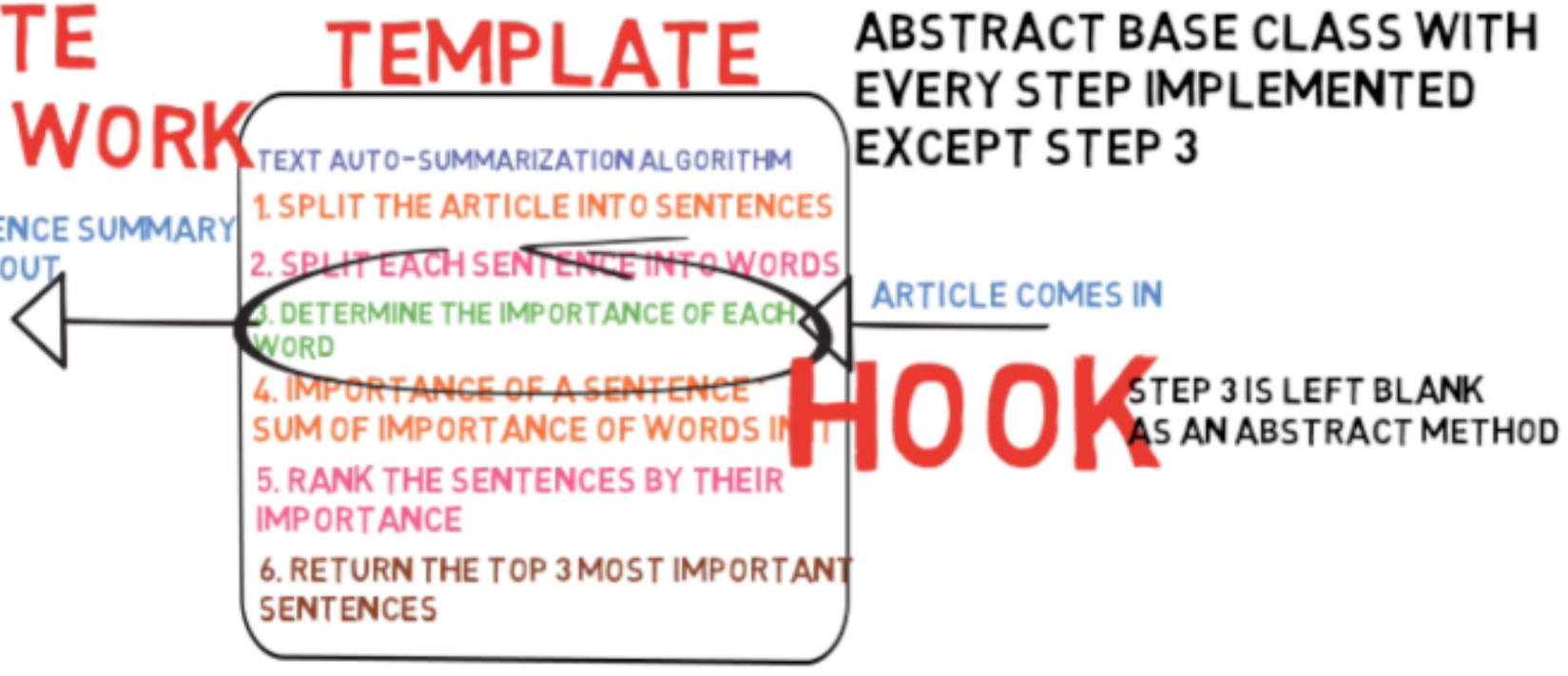
OR SIMPLY PUT -

HIGH-LEVEL CODE SAYS TO LOW-LEVEL CODE: "DON'T CALL US, WE'LL CALL YOU"

FRAMEWORKS ARE, IN FACT, A NATURAL EXTENSION OF

THE TEMPLATE PATTERN

THE TEMPLATE PATTERN AT WORK



ABSTRACT BASE CLASS WITH EVERY STEP IMPLEMENTED EXCEPT STEP 3

STEP 3 IS LEFT BLANK AS AN ABSTRACT METHOD