

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

TEMPLATE

ABSTRACT BASE CLASS WITH
EVERY STEP IMPLEMENTED
EXCEPT STEP 3

3 SENTENCE SUMMARY
COMES OUT

TEXT AUTO-SUMMARIZATION ALGORITHM

1. SPLIT THE ARTICLE INTO SENTENCES
2. SPLIT EACH SENTENCE INTO WORDS
3. DETERMINE THE IMPORTANCE OF EACH WORD
4. IMPORTANCE OF A SENTENCE =
SUM OF IMPORTANCE OF WORDS IN IT
5. RANK THE SENTENCES BY THEIR
IMPORTANCE
6. RETURN THE TOP 3 MOST IMPORTANT
SENTENCES

ARTICLE COMES IN

HOOK

STEP 3 IS LEFT BLANK
AS AN ABSTRACT METHOD