In the problem that we saw before we had a single constructor initializing all the attributes of all three type of products. That makes it difficult to keep track of all the values and if we miss any one of them may result in crashing of the code and high chances of introducing bugs.

Consider if there were more categories of products each having at least 5 attributes it would make the constructor very lengthy.

Constructor Overloading could have been the solution to this problem, but what if no.of attributes of two categories are same also the common attributes like name and price would get repeated every time and this still wont solve the problem of the no.of if-else checks.

Whenever there are a no.of IF-ELSE checks and the condition check is repeating several times we should conclude that we are missing out on something very important that can help us optimize the code.This is one of the very common "ANTI-PATTERNS".

An anti-pattern is a common response to a recurring problem that is usually ineffective and risks being highly counterproductive.

Solution to this is to separate out all the three categories in three different classes that extend a super class that initializes the common attributes and functions for all of the three sub classes.