1. What is the difference between cohesion and coupling?

While cohesion revolves around the portion of a module pertains to a certain purpose (which is perceived as positive in programming), coupling deals with the interdependence of modules in, for instance, accomplishing a goal (which is perceived as negative in programming).

2. Why does modularizing a program make it more efficient?

Modularizing a program leads to a clear structural makeup, where each constituent procedure is divided into executable segments that are easier on the eye than a large mass of code. Debugging issues is also massively simplified, hence further efficiency.

3. What are the advantages and disadvantages of modularization?

The advantages of modularization include complexity that has been abstracted away from and simplified diagnosing of issues. The disadvantages include an unclear understanding of how each code segment reaches a holistic summit (goal) (how each module interacts with the other in order to collectively reach a solution) for old school developers who may not have experience in modularization and overengineering (breaking code into way too many modules, which induces programming complications).