1 What is a result of using the *Aspects* architectural style in the software architecture of a system?

A Does not change the existing modules of the system, because they are determined by the system's decomposition, which is not changed.

B Adds restrictions to the dependency relationships that exist between modules and that are represented using other styles, as with the layered style.

C Introduces only a new type of relation among the existing modules of the system, which resulted from other styles of the module viewtype.

✓ Typically gives rise to more modules than what we would have if not using this style.

2 How can the *Generalization* architectural style of the module viewtype be use to support the evolution of a system?

A By adding, removing, or changing children.

B By changing the commonalities that are in the children.

✓ By changing a parent, which will automatically change all the children that inherit from it.

D Because the *is-a* relation does not allow reuse of implementation.

3 Why does the *Aspects* architectural style promotes the modifiability of a system?

✓ It separates in new modules responsibilities that were spread over various of the system's modules.

B It allows the decomposition of each of the system's modules into finer grained modules.

C It imposes restrictions on which uses relationships may exist between the system's modules.

D It makes it easier to create generalization relationships between the system's modules.

4 In which situations can the *Generalization* style of the Module viewtype be used?

✓ All options are true.

B To describe the extension of a module.

C To express the commonalities of several modules.

D To express module reuse.

5 What is the advantage of using the *Generalization* architectural style in the architecture of a system, when the parent module does not contain an implementation?

A There is no advantage.

✓ It allows to have different implementations of the parent and they can be replaced with little impact on other modules.

C This situation is not possible, because the *Generalization* style requires the parent to contain an implementation.

D All the other options are false.