Home Work 2:

Sabina Akter

7.5 What is the influence of cohesion on maintenance?

Cohesion refers to the relationship within the elements of a module. Modules with high cohesion is apt. Maintenance is easier with these types of cohesion.

-Corrective maintenance is easier in functional cohesion. But in case of logical cohesion, there are severe maintenance problems.

-While doing maintenance in temporal cohesion, there may be lot of modifications to do.

7.6 What is the influence of coupling on maintenance?

Coupling refers to the level of independence between the two software modules.

Coupling is usually contrasted with cohesion. [Low coupling](https://en.wikipedia.org/wiki/Loose_coupling) often correlates with high cohesion, and vice versa. Low coupling is often a sign of a well-structured [computer system](https://en.wikipedia.org/wiki/Computer) and a good design, and when combined with high cohesion, supports the general goals of high readability and maintainability.

7.8 Which of the five levels of coupling described in Section 7.3 promote reuse?

Modules with low coupling is most preferable for reuse. So, data coupling is the most common type of coupling and it is the only one that promotes reuse.

-Because if two modules are data coupled, then maintenance is easier, because there is less chance of inducing regression on one another that promotes reuse

7.10 Distinguish between data encapsulation and abstract data types.

Encapsulation is wrapping, just hiding properties and methods. Encapsulation is used for hide the code and data in a single unit to protect the data from the outside the world. Class cab be the best example of encapsulation. Abstraction is used to hiding something too but in a higher degree (class, interface). Clients use an abstract class (or interface) do not care about who or which it was, they just need to know what it can do.

7.11 Distinguish between abstraction and information hiding.

The difference between abstraction and information hiding is below:

|  |  |
| --- | --- |
| Abstraction | Information |
| The basic of abstraction is extracts only relevant information and ignore inessential details. | The basic of information is hides the data from the parts of the program. |
| The purpose of abstraction is to hide the complexity. | The purpose of information is to achieve encapsulation. |
| Abstraction focusses of observable behavior of the data. | Information focusses on restricting or allowing the use of data within a capsule. |
| Class uses the abstraction to derive a new user-defined datatype. | Information hiding is used in a class to make its data private. |

7.12 Is inheritance a subset of association?

Inheritance:

The procedure of creating new classes from existing classes.

Association:

It is a relationship between two different classes.

As existing class and new class belongs to association and as inheritance is the procedure of creating new classes from existing classes, we can say that inheritance is a subset of association.

7.13 Distinguish between polymorphism and dynamic binding.

Polymorphism:

It refers the technique that can be applied the objects of various classes.

Dynamic Binding:

The act of connecting an object to the proper method at runtime is termed as dynamic binding.

References:

Textbook: Classical Software Engineering

Wikipedia & Class lectures