Module 04 : Object based Decomposition

Notes

Modularity is the process of decomposing a problem (program) into a set of modules so as to reduce the overall complexity of the problem. Booch has defined modularity as – "Modularity is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules." Broadly speaking, modularity is the degree to which a system's components may be separated and recombined, often with the benefit of flexibility and variety in use.

• Decomposing a problem is a done to achieve modularity

Decomposition is the process of breaking a large problem into more manageable subproblems. The motivating principle is that large problems are disproportionately harder to solve than small problems.

• Modularity has its many benefits

The benefits of modular programming are:

Efficient Program Development

Programs can be developed more quickly with the modular approach since small subprograms are easier to understand, design, and test than large programs. With the module inputs and outputs defined, the programmer can supply the needed input and verify the correctness of the module by examining the output. The separate modules are then linked and located by the linker into an absolute executable single program module. Finally, the complete module is tested.

Multiple Use of Subprograms

Code written for one program is often useful in others. Modular programming allows these sections to be saved for future use. Because the code is relocatable, saved modules can be linked to any program which fulfills their input and output requirements. With monolithic programming, such sections of code are buried inside the program and are not so available for use by other programs.

Ease of Debugging and Modifying

Modular programs are generally easier to debug than monolithic programs. Because of the well defined module interfaces of the program, problems can be isolated to specific

modules. Once the faulty module has been identified, fixing the problem is considerably simpler. When a program must be modified, modular programming simplifies the job. You can link new or debugged modules to an existing program with the confidence that the rest of the program will not change.

- Object based decomposition is key to system modularization based on the quality aims of
 - Extendibility
 - Object based decomposition is more effective as objects are much more extendible over time.
 - Additional features can be easily accommodated through instance attributes and behaviors can as well be scaled.

Reusability

- Since object are intrinsically designed as independent components they automatically emphasize reusability.
- Additionally features such as polymorphism and inheritance support reusability.

• Compatibility

- Object based decomposition does have a strong coupling between the different modules in spite of each being independent.
- This promotes seamless inter-working among the various modules.

The advantages of decomposition are -

- The individual components are of lesser complexity, and so more understandable and manageable.
- It enables division of workforce having specialized skills.
- It allows subsystems to be replaced or modified without affecting other subsystems.

The limitations of decomposition are -

- The decomposition process becomes complex, especially for very large systems.
- A complete discourse of the entire system is required to be done meticulously.