[0408] Limitations of Structured Programming

Ultimately, computer programmers discovered that Structured Programming had several weaknesses that prevented them from writing better programs. Many of those weaknesses were addressed in what is now called "Object-Oriented Programming" - which we will look at in detail starting in the next module.

That said, it is important to remember that all object-oriented programs still have - at their core - top-down, structured concepts inside of them. To say that OO programming replaced structured programming is not correct. In many ways, OO is really an extention of structured programming concepts - granted it is a very powerful extension and some OO programs work very differently - but in order to fully understand OO programming (including C# programming) it is important to understand structured programming too.

Limitations

When people started creating really large programs with Top-Down design and structured programming concepts, things went much better than before and they made lots of good progress. Ulitmately however, they ran into roadblocks when it came to the way data was handled. People were no longer hitting bugs in how their programs executed (i.e. because of spaghetti code), instead they were having problems related to when and where data was being modified.

This is because, at its heart, top-down design is all about the flow of control in a program. i.e., what a program does and the order of the statements that get executed. The data that is generated by the code is often an after thought. By over-focusing on process and under-focusing on data, **top-down designs often run into data-oriented issues very late in their development cycle** which can kill the entire project or cause problematic compromises.

Top-Down design also de-emphasizes the reuse and sharing of code modules. Reuse is only looked at during the final phase of a project. Code from other systems is rarely used because the **top-down design usually results in domain-specific requirements for each module**.

While many of these issues can be addressed in a structured-programming project by an experienced programming team, there are fundamental aspects of OO programming that make it a better approach for dealing with those issues. As we will see soon...