

Software Design and Architectures
SE-2 / SE426 / CS446 / ECE426
Fall 2003

Assignment 6 : Modularity in Java or C++

Due in class (1pm RCH 205) on Monday November 10.

Choose to do this for Java or for C++.

In practice modules have to be represented in code and as source files. Propose a conventional representation for modules in your chosen language, to be used in a project such as your telephony project. Define the representation with attention to the following:

coding pattern : what syntax pattern(s) are proposed to represent a module
source filing : what source artefacts (files, directories) and organization is proposed
export : propose a convention for a module to declare its exports
import : propose a convention for a module to declare its imports

Keep your proposal simple and usable. You may find that suitable mechanisms already exist, in which case you can choose to adopt them for your convention.

The proposal for export and import should be (if possible) such that the compiler or linker can be made to check conformance. (For example, the **public** attribute of Java classes, or the declare-before-use requirement of C++ **externs**, are useful to enforce export declarations.) Your import declarations can specify a whole module or only the parts of interest. (Your export declarations must ultimately specify the parts being exported, of course.)

Keep in mind what we say a module is: it must be the right size to be an independent assignment of work to a developer, and suitable for being the boundary of information hiding and good coupling and cohesion practices. It ought to be possible to partition the code of a system into modules: that is, for every line of code in the source we ought to be able to say what module it belongs to.