Drafting a Course Syllabus

These suggestions are phrased in terms of courses, although we are not necessarily working at the level of complete courses. Instead, our focus is on broad topics in scientific computing that students would benefit from knowing about, which could be mixed and matched into different courses that are either entirely computational in focus or they could be integrated into more traditional "methods" and theory classes.

- **Audience and Program**
  - Where does this course fit into the curriculum?
  - What are the prerequisites?
  - What level of students is the course intended for?
  - What majors might take this course?
- **General Course Objectives**
  - What is the theme, or organizing principle, that permeates this syllabus?
  - What are the goals of this course?
  - What are tangibles, broadly described, that you want the student to take away from the course?
  - How will this course achieve these tangibles?
  - How might you convince a student that it is worthwhile to take this class?
    - What are the fundamentals/core of the particular topic which students can leverage in future work in this area?
    - What skills will the student gain practice in?
    - Why is this material relevant, studied? What are further topics students might want to explore? Why is this not a trivial topic that ends at the completion of this course, i.e. what are extensions of this material?
    - What general, critical thinking and other related skills will this course impart that relate to the practice of science discovery?
- **Specific learning outcomes**

  For each of the general course objects, come up with specifics about the particular topics that will be taught within the course and how these relate to the overarching themes of the course that emphasize higher-level, more abstract concepts and thinking.

  Example: A topic in data technologies is relational databases. The specific learning outcomes might be that by the end of the course the student should be able to:

  - Describe the relational model of tables and the algebra of database operations, and explain where and why it is used;
  - Compare the client-server model to other models;
  - Use the syntax and semantics of the query language (SQL) to formulate queries;
  - Understand how SQL is integrated with other languages such as R and Perl;
  - Recognize generally how to approach problems involving data within a database, e.g. determine when to move the data back to R, or do the computations in the database or combine computations in both R and the database.
- **Activities & Involvement**

- **Assignments.** What is the nature of the work expected from the students? Provide examples of assignments and projects that highlight and teach the specifics and also others that get at the broader themes and concepts.
- How are students expected to work on the assignments, e.g. in teams on a project?
- What sorts of pedagogical approaches are most suitable for conveying this material?
- **Materials.** Provide examples of readings - books and papers, Web sites, problems, and projects.

Meta-Example: Example of an example: In "Curriculum Guideline for Bachelor of Arts Degrees in Statistical Science" (Tarpey, Acuna, Cobb, De Veaux, 2001), advocate that the "**approach** to teaching statitistics topics should

- emphasize **real** (not merely realistic) **data** and authentic applications;
- include experience with statistical **computing**, both for data analysis and for simulation or modern computer intensive methods like the bootstrap, using one or more software packages of the sort used by professional statisticians;
- encourage **synthesis** of theory, methods, and applications;
- and offer frequent opportunities to develop **communication** skills through group work, oral presentations, and writing assignments."

Offer details about examples eventhough they will go out of date. Describe how these examples/case studies illustrate the material and are related to the overarching concepts. In other words, relate these examples to the goals of the class.

- **External Justification**
  - Compare this course to other courses that a student might take and therefore explain why it is relevant to our curriculum. The contrast is two-fold: compare with classes on similar topics, e.g. classes taught in CS, and with other statistics classes that focus on entirely different aspects of statistical theory, e.g. wavelets, graphics models, etc.
  - Can we help identify what sub-topics in other classes are optional and can be replaced with computational topics?
  - What are the synergies with other disciplines and how can we make this pertinent to students in other discplines?
  - How can we convince our faculty that this class will draw people in?