Good Course Design and CS Materials

Kalpathi Subramanian[1], Erik Saule[1], Jamie Payton[2], and Matthew Mcquaigue[1]
krs@uncc.edu, esaule@uncc.edu, payton@temple.edu, mmcquaig@uncc.edu

[1]The University of North Carolina at Charlotte
[2]Temple University

SIGCSE 2022

Table of Contents

Curriculum Guidelines

What are they?

Usually they are recommendation of what should/could be taught across a program.
Expressed in term of topics, learning outcome, and competencies. Not in term of courses.
Usually make recommendation on how much one should learn in a particular topic, sometimes specified in number of hours.

How can we use them?

Give us a reference of what we should/could be teaching.
Am I covering all that? Should I? Why not?
Give us a common language to communicate between instructors.

General Guidelines: ACM/IEEE CS 2013

Structured in

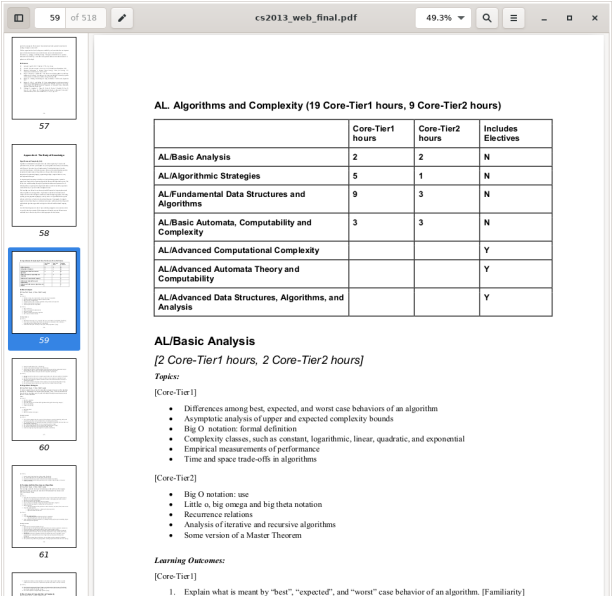
- Knowledge Area
- Knowledge Unit

Topics and Learning Outcomes are classified as

- Tier-1
- Tier-2
- Elective

Other general guidelines:

- Data Science
- Computer Engineering
- Upcoming revised CS



Specific Guidelines: NSF/IEEE-TCPP PDC 2012

Structured in domains:

- Programming
- Algorithm
- Architecture

More descriptive.
Bloom levels.
Other specific guidelines: graphics, security

Activity

- Look at the ACM/IEEE CS 2013 guidelines.
- Find some entries relevant to one of your course.
- But also browse it to get a sense of the scope of it.
- Notice the examplar at the end. Find and read through an examplar for a course similar to what you teach.

Table of Contents

A Lingua Franca

CS Guidelines give us a fairly detailed description of what is in CS.
We can use them as ontologies to describe in a common language what a course of a class material is like.

What do you think is in a lecture entitled UNCC-ITCS-2214-Saule-Graphs?

- Depth- and breadth-first traversals
- Representations of graphs (e.g., adjacency list, adjacency matrix)
- Reflexivity, symmetry, transitivity
- Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree.
- Undirected graphs
- Directed graphs
- Weighted graphs
- Iterative and recursive traversal of data structures

Study of Coverage

We can easily understand what one course is covering.
We can understand across multiple offfering of the same course what that particular course is about.
We can identify different “flavors” of that course.

Activity

Look at the different data structure course using the coverage map.
For a particular course:

- Note something they are teaching and that you were not expecting.
- Note something you thought they would cover and are not covering

Look at all courses at once:

- What are the key topics/outcome that are covered by most?

Table of Contents

What is Alignment?

Properties of how content flow in

- Program
- Course
- Module

That could apply to

- Topics
- Outcomes
- Competencies

That could be in term of

- What they cover
- What they assume students know

Aligning Modules with Course Objectives

Courses usually have objectives that come from program descriptions and assessments.
How do we ensure that the content of the class actually serve these higher objective? We want to align the objective modules with the objective of the course.
Two main properties to check:

- Are all the course objectives covered appropriately by a module objective?
- Are there module objectives that serve no course objective?

Alignment within Module

Typical module structure

- Exposition to new concept (lecture, textbook)
- Clarification of concept (discussion, hands-on activity)
- Reinforcement of concept (problem, programming assignment)

Properties you want

- The clarification should not introduce new concepts
- The reinforcement should strengthen the exposition and clarification topics
- The materials should cover the topics the module is meant to cover
- The materials should not wander too far from the module objectives

Assessment

Exam should never introduced new concepts

Activity

For a particular course, look at the lectures and assignment

- Are there topics in the assignment that are not part of the lecture?
  - Do you think it is a problem?
- Are there topics in the lecture that are not in the assignment?
  - Do you think it is a problem?

Consider two sections of data structures

- Can you identify differences between the two sections
  - Are any of this difference style or fundamental?

Table of Contents

Have you ever searched for materials?

Let's look at Nifty Assignments

Nifty Assignments

The Nifty Assignments session at the annual SIGCSE meeting is all about gathering and distributing great assignment ideas and their materials. For each assignment, the web pages linked below describe the assignment and provides materials -- handouts, starter code, and so on.

Applying for Nifty is now done as its own track with a similar deadline to special sessions. The format and content of the .zip you submit is unchanged. See the info page for ideas about what makes a nifty assignment and how to apply for the Nifty session.

Please email any suggestions or comments to the nifty-admin email: nifty-admin@cs.stanford.edu
Nick's Home

Nifty Assignments 2021

Sankey Diagrams - Ben Stephenson	CS1 Sankey diagram - neat data visualization algorithm
Rocket Landing Simulator - Adrian A. de Freitas and Troy Weingart	CS1 Rocket Landing Simulator - fun algorithm
Covid Simulator - Steve Bitner	CS1-CS2 Covid 2D infection simulator - timely if scary
Linked List Labyrinth - Keith Schwarz	CS2 Neat memory / debugger skill exercise, custom per student

Nifty Assignments 2020

Thanks to our presenters for getting everything together including videos for this COVID-interrupted year.

Typing Test - John DeNero et al	CS1 Fill in algorithm of fun typing-speed test. **(Video)** (intentionally silent)
Color My World - Carl Albing	CS1 or later: Students are given a data file, but no description about what it represents. Can they solve the mystery by generating a reasonable image?
Bar Chart Racer - Kevin Wayne	CS1 - use real data to make a animated bar chart - captivating! **(Video)**
DNA - Brian Yu, David J. Malan	CS1 or CS2 Neat DNA project. **(Video)**
Recursion to the Rescue - Keith Schwarz	Nifty recursion projects using tied to real-world applications. **(Video)**
Decision Makers - Evan Peck	Two hour exercise illuminating algorithms and life

Nifty Assignments 2019

Nifty Post It - Jeffrey L. Popyack	CS0-CS1 Hands On Manipulative
Hawaiin Phonetic Generator - Kendall Bingham	CS1 Fun Text
Motion Parallax - Ben Dicken	CS1 Awesome Graphic Experience

Metadata

Summary	Students develop a program to map raw data files into a colorful images.
Topics	visualization, big data, image processing - color maps.
Audience	Use as an early assignment in an HPC class, Scientific Programming class, Data Science/Analysis class, or a Graphics/Image processing class. Appropriate for CS1 or higher students familar with loops, file io, argument parsing, and image processing. The starter code is written in Python.
Difficulty	This assignment is appropriate for various levels, depending on the initial conditions: starter code (or not), existing color maps (or not) and time alloted. A late-semester CS1 class given the starter code and a week.

Strengths	• Solving the mystery of what the image "looks" like
	• Working with *real-world* data to get visual, graphical feedback.
	• Allows for some artistic flair resulting in variations among solutions
	• Depending on the assignment write up there are open ended options including:
	  ○ creating different colormaps for different images;
	  ○ scaling the data to fit a given image size;
	  ○ a "smarter" program to deduce the image size from the data file;
	  ○ statistical analysis of the data to drive the choice of color map values

Weaknesses	• When creating a colormap from scratch it can be tricky to get color assignments that are both visually pleasing (artistic) and pull out the desired details, though that is part of the point of this assignment. • Use of graphics makes unit testing more challenging.
Dependencies	

Curriculum Guidelines as Features

Features

The problem in classic search is that it is hard to find good matches because people use imprecise textual descriptions.
Curriculum guidelines give us a well established precise features

Search

Give a set of materials that use these topics/outcomes

Recommendation

Give a set of materials that match the same outcomes as these ones.

Activity

Can you find materials about hash tables?
Can you find materials about shortest path?