

CS Materials: A System to Assess and Align Your Courses to National Standards

Alec Goncharow⁺, Matthew McQuaigue⁺, Erik Saule⁺, Kalpathi Subramanian⁺,
Jamie Payton*

⁺Dept. of Computer Science, University of North Carolina at Charlotte

*Dept. of Computer & Information Science, Temple University

esaule@uncc.edu

This material is based upon support from NSF OAC 1924057, DUE-1726809 and CCF-1652442.

Agenda

- 1 Introductions (10 min)
- 2 What is CS Materials and how can it help? (35 min)
- 3 Playground (15 min)
- 4 Discussions (20 min)
- 5 Wrap up (10 min)

Table of Contents

- 1 Introductions (10 min)
- 2 What is CS Materials and how can it help? (35 min)
 - In Spirit
 - CS Materials
 - As a repository
 - As a tool on a class
- 3 Playground (15 min)
- 4 Discussions (20 min)
- 5 Wrap up (10 min)

Introductions

- Who are we?
- Who are you?

Table of Contents

- 1 Introductions (10 min)
- 2 What is CS Materials and how can it help? (35 min)
 - In Spirit
 - CS Materials
 - As a repository
 - As a tool on a class
- 3 Playground (15 min)
- 4 Discussions (20 min)
- 5 Wrap up (10 min)

(Re)Designing courses for the modern era is involved

Learning objective first

A national effort by educators to redesign courses in term of learning objectives first than in term of a linear set of topics.

(Re)Designing courses for the modern era is involved

Learning objective first

A national effort by educators to redesign courses in term of learning objectives first than in term of a linear set of topics.

Alignment efforts

SACS, ABET, Quality Matters to reach better alignment.

- Program objectives with course objectives.
- Course objectives with module objectives.
- Module Objectives with Content (lectures, assignments, exams).
- Student centered, student success.

(Re)Designing courses for the modern era is involved

Learning objective first

A national effort by educators to redesign courses in term of learning objectives first than in term of a linear set of topics.

Alignment efforts

SACS, ABET, Quality Matters to reach better alignment.

- Program objectives with course objectives.
- Course objectives with module objectives.
- Module Objectives with Content (lectures, assignments, exams).
- Student centered, student success.

Course design questions

- Is what I cover reasonable?
- What to expect from incoming students?
- What are my colleagues doing?
- What do follow-on courses expect?

Following curriculum recommendations requires expertise!

Integration of modern topics/objectives

With the field changing, new topics require integration in curriculum

- Computer Security
- Ethics in Computing
- Parallel and Distributed Computing

Following curriculum recommendations requires expertise!

Integration of modern topics/objectives

With the field changing, new topics require integration in curriculum

- Computer Security
- Ethics in Computing
- Parallel and Distributed Computing

Splice content into courses is the right approach, but overwhelming

Take integrating PDC into Early CS Courses as an example:

- Cell phones are multicore.
- Game consoles have GPUs.
- Science runs on clusters.
- Data analysis runs in reconfigurable data centers.
- Home's IoT are mesh networks.
- Websites are distributed services.
- Data transfers are peer to peer.
- Money becomes decentralized.

Following curriculum recommendations requires expertise!

Integration of modern topics/objectives

With the field changing, new topics require integration in curriculum

- Computer Security
- Ethics in Computing
- Parallel and Distributed Computing

Splice content into courses is the right approach, but overwhelming

Take integrating PDC into Early CS Courses as an example:

- Cell phones are multicore.
- Game consoles have GPUs.
- Science runs on clusters.
- Data analysis runs in reconfigurable data centers.
- Home's IoT are mesh networks.
- Websites are distributed services.
- Data transfers are peer to peer.
- Money becomes decentralized.

If you have questions on how to integrate PDC in Early CS, ask me offline.

The difficulty of reaching national change

We Need a Better Understanding of How CS is Taught!

- What topics are taught in early CS courses?
- What is the sequence of topics?
- Programming languages used: OOP vs Procedural vs Functional.
- Course sequences and their implications for success.
- Student preparation, diversity, pedagogies, engagement.
- Can we get a better understanding of all of these and relate them to student success?

CS Materials can be a solution to all these problems

CS Materials will help you

- Check the coverage of a course.
- Study alignment of content and assessment.
- Compare your section to other sections.
- Relate a course with prerequisite and follow on.
- Identify what materials need to be built.
- Find new materials when you are designing a course.
- Share your cool materials.

What is CS Materials?

On the surface, CS Materials is an actively managed repository

- A repository of materials (lecture slides, videos, assignments, exams).
- Mapped (and continually updated) to current curriculum guidelines:
 - ▶ ACM CS 13
 - ▶ NSF/IEEE-TCPP PDC 12
 - ▶ more to come
- With search functions.
 - ▶ An *Amazon* for materials.
- Recommendation features.
 - ▶ Based on similarity with some of your content.

What is CS Materials?

On the surface, CS Materials is an actively managed repository

In depth, CS Materials is a tool to understand courses

- to classify materials.
 - ▶ an instructor can gain insight about his/her own class.
- to inspect the coverage of a class or a set of materials.
 - ▶ what is covered.
 - ▶ what is not covered.
- to study alignment of a class.
 - ▶ align content with assessment.
 - ▶ align one course with the next.
 - ▶ align your section with a colleague's section of same course.
- to understand how a class is being taught across different universities and programs.

What is CS Materials?

On the surface, CS Materials is an actively managed repository

In depth, CS Materials is a tool to understand courses

In reality, CS Materials is a still-growing system

- There are few materials currently in the system.
- The project is a community effort to build up the system.
- We will develop the tools to leverage the system based on user feedback.

Curriculum guidelines as lingua franca for CS education

Curriculum guidelines

- By a diverse committee of stake holders.
- Describes:
 - ▶ what must be covered.
 - ▶ what should be covered.
 - ▶ what could be covered.
- Often gives estimates of number of hours required.
- Typically organized as a tree.

There is a guideline for your field

ACM/IEEE CS 2013, CE 2016, CSEC 2017, IS 2010, IT 2017, SE 2014; **NSF/IEEE-TCPP PDC 2012**; College board AP CS A, AP CSP; NSA COF; SIGGRAPH EC CGT 2001

AL. Algorithms and Complexity (19 Core-Tier1 hours, 9 Core-Tier2)

	Core-Tier1 hours	Core-Tier2 hours
AL/Basic Analysis	2	2
AL/Algorithmic Strategies	5	1
AL/Fundamental Data Structures and Algorithms	9	3
AL/Basic Automata, Computability and Complexity	3	3
AL/Advanced Computational Complexity		
AL/Advanced Automata Theory and Computability		
AL/Advanced Data Structures, Algorithms, and Analysis		

AL/Basic Analysis

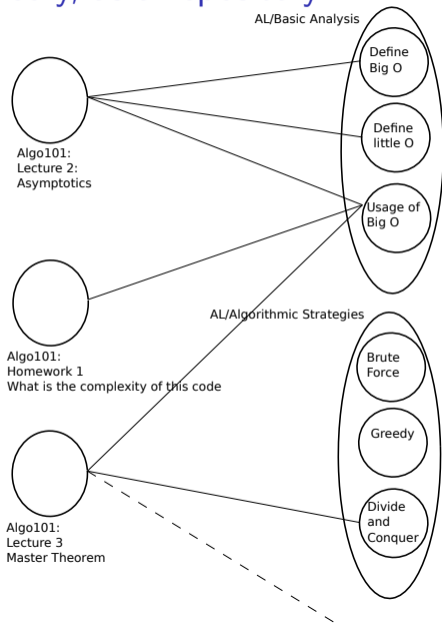
[2 Core-Tier1 hours, 2 Core-Tier2 hours]

Topics:

[Core-Tier1]

- Differences among best, expected, and worst case behaviors of an algorithm
- Asymptotic analysis of upper and expected complexity bounds
- Big O notation: formal definition
- Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential
- Empirical measurements of performance
- Time and space trade-offs in algorithms

Theory, as a repository



Depending on your philosophical bias, you can see classification as:

- a bipartite graph.
- a hypergraph.
- 0-1 matrix.

Searching is a similarity search in graph.
Recommending is a classic problem.

Demo

- Inputting a material `https://cs-materials.herokuapp.com/material/142/edit`
- Making a Collection

Harmonizing classification of a class

- There are so many classification items it is likely the classification of your materials would have some inconsistencies.
- What you want is to see a set of materials at once to compare classifications holistically.
- Identifying rows and columns commonly correlated helps presenting the information concisely. (That's a biclustering problem.)

Demo: <https://cs-materials.herokuapp.com/matrix?ids=178>

Coverage Analysis

Questions

- What is my course covering?
- Are there topics I ramble on?
- Are there topics I should cover but don't?
- Are there topics for which we don't have good materials for?

Demo: <https://cs-materials.herokuapp.com/radial?ids=178>

Studying alignment of your class

Questions

- Are the topics of the exam covered in my lectures?
- Am I covering the class the same way my colleagues are? <https://cs-materials.herokuapp.com/radial?tree=acm&listoneids=178&listtwooids=185>
- Are my lecture and homework aligned? <https://cs-materials.herokuapp.com/radial?tree=acm&listoneids=182&listtwooids=181>

Search and Similarity

- Searching for new materials. Search for “define Big O” at <https://cs-materials.herokuapp.com/materials>
- Soon to come: Search by Similarity (“you may also be interested...”)

Some longer term plans/opportunity

Simplifying Classification

With more data, one can start applying ML techniques to simplify input.

Some longer term plans/opportunity

Simplifying Classification

Integrate concepts of engagement

This could give equivalent assignments that could not only be more relevant but also engaging to your students.

Some longer term plans/opportunity

Simplifying Classification

Integrate concepts of engagement

Understanding flavor of classes

There are dozens of ways of teaching CS1, we could map the main approaches, explore/compare ways that can lead to student success.

Some longer term plans/opportunity

Simplifying Classification

Integrate concepts of engagement

Understanding flavor of classes

How do topics/classes flow into one another

We tend to have a static view of how to teach CS, but there may be better ways. Let's understand the program/degree as a whole (of great use to program designers/administrators).

Some longer term plans/opportunity

Simplifying Classification

Integrate concepts of engagement

Understanding flavor of classes

How do topics/classes flow into one another

Degree Audit

Do all students actually cover all the topics that are recommended?

Some longer term plans/opportunity

Simplifying Classification

Integrate concepts of engagement

Understanding flavor of classes

How do topics/classes flow into one another

Degree Audit

Supporting evolving standards

Adding new standards is not much work. And mapping a standard to its next iteration can be done semi automatically

What is in the system right now?

- About 50% of Nifty Assignment
- All Peachy parallel assignment before 2020
- Saule Parallel Computing (UNC Charlotte)
- Saule Data Structure (UNC Charlotte)
- Subramanian Data Structure (UNC Charlotte)
- Subramanian OOP (UNC Charlotte)
- Subramanian Algorithm (UNC Charlotte)
- Payton Capstone (Temple University)
- Glebova Algorithm Analysis (Georgia State University)
- Kerney CS1 (Clovis Community College)
- Kerney CS2/DS (Clovis Community College)
- Wagner DSA (Birmingham-Southern College)
- Duke CS2/DS (Virginia Commonwealth University)
- Wahl Algorithm (Hanover)

Table of Contents

- 1 Introductions (10 min)
- 2 What is CS Materials and how can it help? (35 min)
 - In Spirit
 - CS Materials
 - As a repository
 - As a tool on a class
- 3 Playground (15 min)
- 4 Discussions (20 min)
- 5 Wrap up (10 min)

Playground

Some ideas:

- Coverage of Wahl's Algorithm course
- Could compare Bill Kerney and Debra Duke CS2/DS.
- Could compare Wagner DSA to Wahl Algorithm.
- Could compare Duke CS2/DS to Subramanian DS.
- Search for Hashing content.

Table of Contents

- 1 Introductions (10 min)
- 2 What is CS Materials and how can it help? (35 min)
 - In Spirit
 - CS Materials
 - As a repository
 - As a tool on a class
- 3 Playground (15 min)
- 4 Discussions (20 min)
- 5 Wrap up (10 min)

Discussions

Table of Contents

- 1 Introductions (10 min)
- 2 What is CS Materials and how can it help? (35 min)
 - In Spirit
 - CS Materials
 - As a repository
 - As a tool on a class
- 3 Playground (15 min)
- 4 Discussions (20 min)
- 5 **Wrap up (10 min)**

Surveys

CSMaterials Survey: https://uncc.qualtrics.com/jfe/form/SV_b2B64uVdFPbSky9

Join the CS Materials effort!

Assess your class

Is your class well designed?

Are your lecture and assignment well aligned?

Find out using CS Materials!

Assess your program

Contribute to CS education research

Distribute your good stuff!

Join the CS Materials effort!

Assess your class

Distribute your good stuff!

You have been building cool video for your async class? You have been building engaging assignment?

Who's goign to find them? CS Materials users would!

Help your broader impact using CS Materials

Assess your program

Contribute to CS education research

Join the CS Materials effort!

Assess your class

Distribute your good stuff!

Assess your program

Are you a chair interested in improving your program? Are you designing a new certificate?

CS Materials can help you build a program that is aligned with your goals by construction.

Contribute to CS education research

Join the CS Materials effort!

Assess your class

Distribute your good stuff!

Assess your program

Contribute to CS education research

Help us map how CS is being taught world wide!

Help us find out alternative ways to teach your classes!

Join the CS Materials effort!

Assess your class

Assess your program

Distribute your good stuff!

Contribute to CS education research

Also, we offer stipends for adopters!

Thank you!

Contact:

- esaule@uncc.edu
- krs@uncc.edu
- payton@temple.edu

This material is based upon support from NSF OAC 1924057, DUE-1726809 and CCF-1652442.