



**Coversheet - Proposal for New and Revised Courses**  
(Use for non-Pathways courses)

For CLE/Pathways courses, form can be found here: <https://www.pathways.prov.vt.edu/proposal-forms.html>

**General Information**

<i>Proposal Date:</i>	Fall 2023	<i>Department:</i>	Computer Science
<i>Course Designator and Number (Cross-listed Course Designator and Number):</i>		CS 5020	
<i>Title of Course:</i>	Software Design and Data Structures	<i>Credit Hours:</i>	3
<i>Course Transcript (ADP) Title (30 Characters &amp; Spaces Maximum):</i>		Software Des & Data Structures	
<i>Instructor and/or Departmental Contact:</i>		Trey Mayo - Director of Graduate Programs	
<i>Contact Phone:</i>	(540) 231-0780	<i>Contact E-mail:</i>	treymayo@vt.edu
<i>Please refer to Office of University Registrar for guidelines and policy requirements: <a href="https://registrar.vt.edu/governance.html">https://registrar.vt.edu/governance.html</a></i>			

**Please count this course toward the following Scorecard Metrics areas:**

- Study Abroad       Service Learning       Experiential       Undergraduate Research

Scorecard Metrics Definitions can be found here: <https://registrar.vt.edu/faculty-toolbox/scorecard-metrics.html>

**Please insert an X if this course should count toward First Year Experience:**

- First Year Experience (FYE) Include approval letter from FYE Director. More information can be found here: <http://www.fye.vt.edu>

**Select ONE of the following boxes**

<input checked="" type="checkbox"/> New Course	<input type="checkbox"/> *Revised Course (Revision > 20% _____ Revision < 20% _____)
--	--

For CLE/Pathways courses, form can be found here: <https://www.pathways.prov.vt.edu/proposal-forms.html>

**\*Please include a summary of course revisions to the Justification section of proposal**

<b>A:</b>	<i>Attach statement from Dean or Departmental Representative as to whether teaching this course will require or generate the need for additional departmental resources.</i>		
<b>B:</b>	<i>Attach appropriate letters of support (e.g., prerequisite, corequisite, or cross-list memo) from affected departments and/or colleges.</i>		
<b>C:</b>	<i>Effective Semester:</i>	Fall 2023	
<b>D:</b>	<i>Change in Title From:</i>		
	<i>To:</i>		
<b>E:</b>	<i>Change in Transcript Title (ADP) From:</i>	<i>To:</i>	
<b>F:</b>	<i>Change in Credit Hours From:</i>	<i>To:</i>	
<b>G:</b>	<i>Change in Lecture and/or Lab Hours From:</i>	<i>To:</i>	
<b>H:</b>	<i>Course Number(s) and Title(s) to be deleted from the Catalog with <u>APPROVAL</u>:</i>		

**Approval Signatures**

<i>Department Representative</i>	DocuSigned by: Trey Mayo 8E7851C75088405...	<i>Date</i>	9/22/2022
<i>College Curriculum Committee Rep</i>		<i>Date</i>	9/27/2022
<i>College Dean or Designee</i>		<i>Date</i>	9/27/2022

## Course Information

### Catalog Description

A programming-intensive exploration of software design concepts and implementation techniques. Builds on knowledge of fundamental object-oriented programming. Advanced object-oriented software design, algorithm development and analysis, and classic data structures. Includes a team-based software project. Not for graduate credit for those in Computer Science and Applications. Pre: Graduate standing (3H, 3C).

### Learning Objectives

Having successfully completed this course, the student will be able to:

1. Design data structures such as linked lists, stacks, and queues.
2. Employ an Integrated Development Environment to iteratively develop and test medium-sized programs (e.g., 1K-3K lines of code), including interactive applications.
3. Develop reusable, type-safe software components.
4. Solve problems using recursive algorithms such as for implementing a binary tree.
5. Apply basic algorithm analysis to simple algorithms.
6. Apply software engineering approaches to team-based development.
7. Logically defend positions on ethical issues arising in the design and engineering of software solutions.

### Justification

This course is proposed to help graduate students improve their software design and development proficiency. This includes students who seek to enter a Computer Science and Applications degree but need to make up for deficiencies they have at the time of application, students seeking to later take CS 5040 but who do not have the requisite background, or other graduate students seeking to improve their programming skills. This course will not require any additional resources to teach.

This course is offered at the 5000 level because it provides the necessary skills and knowledge to graduate students to help improve their software design and development proficiency they may not have received in their undergraduate training. It is expected that students will have some foundational knowledge in object-oriented programming and basic software engineering principles.

### Prerequisites and Corequisites

Pre: Graduate standing



Texts and Special Teaching Aids

Required: None. No text is required.

The course will utilize material including approximately 120 course videos (ranging from 3-15 minutes each), course readings, and interactive coding exercises in addition to quizzes and projects. The recommended text is supplemental, and the course materials used are aligned well with the recommended text.

Recommended: Carrano, F. M., & Henry, T. (2014). *Data structures and abstractions with Java* (4th ed). Pearson. pp. 936

Examples of Articles:

- Association of Computing Machinery. (2018). *ACM Code of Ethics and Professional Conduct*. Association of Computing Machinery. <https://www.acm.org/code-of-ethics>
- Dear Data. (n.d.). *Dear data: The project*. Dear Data. <http://www.dear-data.com/theproject>
- OpenDSA. (2022). *OpenDSA: CS 5020 Software Design & Data Structures*. OpenDSA. <https://opensa-server.cs.vt.edu/OpenDSA/Books/CS5020/html/#>
- Oppel, Jr., R. A., et al. (2020, July 5). *The fullest look yet at the racial inequality of the coronavirus*. The New York Times. <https://www.nytimes.com/interactive/2020/07/05/us/coronavirus-latinos-african-americans-cdc-data.html>
- The Madison Collaborative. (n.d.). *The eight key questions handbook*. James Madison University. <https://www.jmu.edu/ethicalreasoning/docs/131101%208KQ%20Handout%20Revision.pdf>

Topic Syllabus

<u>Topic</u>	<u>Percent of Course</u>
Object-oriented software design	25%
Organizing, managing and creating multi-class designs	
Inheritance, abstract classes, polymorphism	
Interfaces and alternative implementations	
Exceptions and Generics	
Allocating responsibilities	
Basic design patterns	
Modeling tools	
Data structures	20%
Design, implementation, and use of classic data structures	
Stacks, queues, lists, binary trees, binary search trees	
Comparisons of implementations	
Algorithms and problem-solving	15%
Algorithm design techniques	
Linked structures	
Using API classes	
Introduction to algorithm analysis	10%
Upper and average complexity bounds	
Big “O” notation	
Searching and sorting	5%
Sequential and binary search	
Simple sorting algorithms	
Recursion	10%
Software engineering issues	15%
Software project management	
Object-oriented testing	
Team-based development	
	Total: 100%



Old (Current) Topic Syllabus

N/A



**Department of Computer  
Science**  
620 Drillfield Drive  
Torgesen Hall, Suite 3210  
Blacksburg, Virginia 24061  
P: (540) 231-0780  
treymayo@vt.edu

September 22, 2022

To: Course Approval Committees

From: Trey Mayo   
Director of Graduate Programs for Computer Science

Re: CS 5020

The Department of Computer Science is requesting approval for a new course proposal for CS 5020  
“Software Design and Data Structures.”

No additional resources will be required in order to offer this course.