## Syllabus for Computer Science 241

***Data Structures and Abstraction***

***Fall 2018***

**Instructor:** Dr. Randy L. Ribler

**Office:** 103 Hobbs, **Phone:** 544-8529

**Class Web Page:** <https://faculty.lynchburg.edu/ribler_r/cs241>

**email:** [ribler@lynchburg.edu](mailto:ribler@lynchburg.edu)

**Office Hours:** will be posted at <https://faculty.lynchburg.edu/ribler_r>

**Course Description:** CS241/242 combine a study of data structures and algorithms in computer systems. Topics may include, queues, stacks, trees, hashing, sorting, graphs, algorithm analysis, and algorithmic problem solving

**Purpose:**

CS241 teaches specific data structures and algorithms and demonstrates methods for creating and evaluating new algorithms.

**Prerequisite for CS242:** CS241

**Course Objectives/Learning Outcomes:**

Students will be able to describe and implement a number of standard data structures and algorithms.

Students will be able to identify the correct data structures and algorithms to apply to specific problems.

Students will be able to evaluate the efficiency of an algorithm in both time and memory utilization.

Students will be able to recognize frequently applied tactics used in the development of efficient algorithms.

Students will be able to apply recursion.

Students will be able to describe and implement algorithms on stacks, queues, lists, strings, trees and graphs.

Students will be able to describe and implement dynamic programming.

Students will be able to describe a number of sorting algorithms and compare their strengths and weaknesses.

**Required Text:**

Algorithms, Fourth Edition, Robert Sedgewick and Kevin Wayne

**Course Requirements:**

**Attendance:** The student is expected to attend all classes and is responsible for all material presented in class. Some material presented in class will be available nowhere else. Past experience indicates that good attendance is an important factor in achieving successful completion of the course.

**Late Policy:** Assignments should be submitted by 11:59pm on the day that they are due. Late assignments will be penalized 5% per day up to a maximum of 40% points. After two weeks, late assignments will not be accepted.

**Plagiarism:** All work must be your own. While it is fine to ask each other questions concerning general concepts, you may not show each other your programs, incorporate code from programs available on the Internet, or exchange any program fragments -- not one line of code. If you are having problems completing your assignments you are encouraged to meet with the instructor.

**Evaluation Methods:**

**Computer Programs (40%)**

**Exams (50%)**

Exam 1 (15%)

Exam 2 (15%)

Comprehensive Final Exam (20%)

**Class Participation and Homework (10%)**

**Computation of Final Grade:**

A z-score will be computed for each assignment. The final grade will be determined from the weighted average of the grades from all assignments. (see Virginia Tech Testing [Memos #6](http://ribler_r.web.lynchburg.edu/VirginiaTechTestingMemos/Memo6-averaged.docx).) However, no student who fails the final exam or leaves two or more programming assignments uncompleted will earn a grade higher than D.

**Center for Accessibility and Disability Resources:**

University of Lynchburg is committed to providing all students equal access to learning opportunities. The Center for Accessibility and Disability Services (CADR) works with eligible students with disabilities (medical, physical, mental health and cognitive) to make arrangements for appropriate, reasonable accommodations. Students registered with CADR who receive approved accommodations **are required to provide letters of accommodation each semester to each professor. A meeting to discuss accommodations the student wishes to implement in individual courses is strongly suggested.** *Accommodations are not retroactive and begin when the accommodation letter is provided to faculty*. For information about requesting accommodations, please visit <https://www.lynchburg.edu/academics/disability-services/> (rev 5/22/18)

Contacts:

Julia Timmons, Accessibility and Disability Resources Coordinator

Email timmons.j@lynchburg.edu, phone (434)-544-8687 or

Meg Dillon, Accessibility and Disability Resources Specialist

Email dillon\_ma@lynchburg.edu, phone (434)-544-8709.

**Inquire**: frame questions that address issues and uncertainties across a range of disciplines.

The student will

* + recognize precise and complete statements of problems.
  + recognize what information is necessary in order to solve given problems.
  + ask essential questions about given problems.
  + ask questions for further study regarding problems and reading assignments.
  + develop an approach for investigating a significant  project.

**Explore:** investigate issues in depth and detail.  
The student will

* + think creatively about possible solutions to problems.
  + use data collection tools, laboratory computer systems, and simulations to gain insights into network protocol complexities.
  + comprehend given problems, reading assignments, and the arguments of others.
  + investigate a significant project independently.
* **Conclude**: develop informed responses to issues.  
  The student will
  + marshal evidence in support of a solution to a problem or conclusion in an argument.
  + articulate an appropriate conclusion based on the evidence.
* **Persuade**: convince others of the validity and value of conclusions.  
  The student will
  + produce precise and complete statements of solutions to problems.
  + construct effective written arguments based in evidence, reason and understanding.
  + deliver effective oral arguments based in evidence, reason and understanding.
* **Engage:** use knowledge and abilities for the good of self and society.  
  The student will
  + demonstrate the ability to project how the use of technologies might work toward the good or detriment of society.
  + value achievements in Computer Science for their intrinsic worth.
  + work effectively with other members of a group to solve problems and present their solutions.