## Syllabus for Computer Science 142

### Introduction to Computer Programming II

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

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

**Class Web Page:** http://ribler\_r.web.lynchburg.edu/cs142

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

**Class Meetings:** MWF 10:00-10:50 (Hobbs 113)

Th 3:00-5:00 (Hobbs 113)

**Office Hours:** M-F 9:00-10:00am or by appointment

**Objective**: This course is the second in a two part series that will provide the student with basic competency in the C++ programming language.

**Text:** C++ Primer, Fifth Edition, by Stanley B. Lippman, Josée Lajoie, Barbara E. Moo

**Software:** Students taking computer science classes at Lynchburg College may download Microsoft development tools, such as Visual Studio, via the Microsoft Imagine [website](https://e5.onthehub.com/WebStore/ProductsByMajorVersionList.aspx?ws=447d0252-c79b-e011-969d-0030487d8897&vsro=8) . Please send me an email if you have a problem with your account.

**Principal Topics:**

Pointers and References

Memory Allocation/Deallocation

Class construction and destruction.

Overloaded Functions

Virtual Functions

Operators

Constness

Namespaces

Templates

Exceptions

Classes

Inheritance

Abstraction

Object-oriented Programming

Standard Template Library

**Grading:**

**5-10 Computer Programs/Labs** (50%) \*

\* Some programs may be worth more than others.

The *interactive evaluation* is conducted during a 15-minute meeting with the instructor. The student will be asked questions about the program’s design and implementation, and will receive guidance in ways to improve. The interactive evaluation will be scheduled periodically. Students must meet with the instructor during their assigned times regardless of the state of their project’s completion. Students who do not attend their interactive evaluation forfeit their ability to submit a late assignment.

**Class Participation** (10%) Class participation includes engaging in class discussions, answering questions, and working on in-class lab assignments. Failure to follow the class conduct rules will result in the loss of some, or all, of your class participation grade.

**Exam 1** (10%)

**Exam 2** (10%)

**Final Exam** (20%) – Saturday, May 13, 2017 at 8:00am

All tests will be closed book and closed notes.

**Computation of Final Grade:**

Z-scores will be computed for each assignment. The final grade will be determined by the weighted average of the grades from all assignments. However, no student who fails the final exam or completes fewer than 3/4 of the labs will receive a passing grade. For details on the grading process, see <http://ericae.net/digests/tm9505>.

**Late Policy:** Assignments should be submitted by 11:59pm on the day that they are due. Late assignments will be penalized 5 points per day up to a maximum of 30 points. Assignments submitted more than two weeks late will receive failing grades. Students who do not attend their interactive evaluation forfeit their ability to submit a late assignment.

**Tutoring:**

The computer science department employs tutors who can be consulted for help with class assignments. Student tutoring is provided at no charge to the student.

**Student Accommodations:**

Lynchburg College is committed to providing all students equal access to learning opportunities. The Disability Services Coordinator (DSC) works with eligible students with disabilities (medical, physical, mental health and cognitive) to make arrangements for appropriate, reasonable accommodations. Students registered with the DSC 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.*** For information about requesting accommodations, please visit <http://www.lynchburg.edu/disability-services>, or contact Julia Timmons, [timmons.j@lynchburg.edu](mailto:timmons.j@lynchburg.edu), phone (434)-544-8687. (rev 5/4/16)

**Attendance and Tardiness:**

Classes will start promptly at the start of the hour. Students who arrive late to class will forfeit their opportunity to take the day’s quiz, and will receive a failing grade for the quiz. Students are expected to attend all classes, and are responsible for the material covered. Past experience has shown that students who regularly miss class have little chance of passing the course.

**Class Conduct:** Students are expected to pay attention in class and respect the basic courtesies commonly found in civilized societies. You will be expected to arrive on time and to stay until the completion of the class. Computers and personal electronic devices should be used for relevant class work only. Emailing, web surfing, playing computer games, and other similar activities are prohibited.

**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 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 or class tutor.

**Objectives:**

Students will be able to write programs utilizing the following programming constructs:

Classes

Templates

Overloaded functions

Class operators

Virtual functions

Inheritance

Exceptions

Namespaces

Students will understand basic object-oriented design and development techniques.

Students will understand how to apply the basic relationships between classes.

Students will be able to incorporate basic constructs from the Standard Template Library.

Students will be understand the concepts of information hiding, encapsulation, and polymorphism and will demonstrate their use in programming assignments.

Students will understand how to employ coding standards to make their programs easy to modify and maintain.

Students will understand constructors, copy constructors, and destructors and will be able to explain how and when these functions execute.

Students will demonstrate these abilities through the completion of programming assignments, in-class examinations, and quizzes.

Students will:

* **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,
  + and develop an approach for investigating program requirements.
* **Explore:** investigate issues in depth and detail. The student will
  + think creatively about possible solutions to problems,
  + use data debugging techniques to understand how their programs are performing,
  + and comprehend given problems, reading assignments, and the arguments of others.
* **Conclude**: develop informed responses to issues. The student will
  + identify program defects/bugs and determine their causes and solutions,
  + and articulate the cause of the defect.
* **Persuade**: convince others of the validity and value of conclusions. The student will
  + show how one approach to a program/problem is better than another,
  + and construct effective arguments based in evidence, reason and understanding.
* **Engage:** use knowledge and abilities for the good of self and society. The student will
  + work effectively with other members of a group to solve problems and present their solutions.