USC Viterbi
School of Engineering

CSCI 102: Fundamentals of Computation (Spring 2020)

Units:	2
Instructor:	Mohammad Reza Rajati, PhD PHE 412 rajati@usc.edu – Include CSCI 102 in subject
Office Hours:	Wednesday 1:30 pm –3:00 pm
TA(s):	Nathan Bartley nbartley@usc.edu – Include CSCI 102 in subject
Office Hours:	TBD
TA(s):	Giovanni Sutanto gsutanto@usc.edu – Include CSCI 102 in subject
Office Hours:	TBD
Course Producers:	TBD tbd@usc.edu – Include CSCI 102 in subject
Lecture:	Section 1: Tuesday, Thursday, 2:00 pm –2:50 pm in ZHS 352 Section 2: Tuesday, Thursday, 3:00 pm –3:50 pm in ZHS 352
Webpages:	Piazza Class Page for discussion and supplementary material and USC Blackboard Class Page for grades and ZyBooks for lectures, code submission, and grades
Prerequisite:	None.
Recommended Preparation:	Proficiency in high school math (including trigonometry, algebra, and basic probability).
Hardware/Software Required:	A laptop and Internet connection will generally suffice to complete homework.
Tentative Grading:	Programming Assignments 30% Labs 5% Class Participation Activities 5% Midterm Exam 30% Final Exam 30% Participation on Piazza* 5%

Letter Grade Distribution:

$\geq$ 93.00	A	73.00 - 76.99	C
90.00 - 92.99	A-	70.00 - 72.99	C-
87.00 - 89.99	B+	67.00 - 69.99	D+
83.00 - 86.99	B	63.00 - 66.99	D
80.00 - 82.99	B-	60.00 - 62.99	D-
77.00 - 79.99	C+	$\leq$ 59.99	F

Disclaimer: Although the instructor does not expect this syllabus to drastically change, he reserves every right to change this syllabus any time in the semester.

Note on e-mail vs. Piazza: If you have a question about the material or logistics of the class and wish to ask it electronically, please post it on the piazza page (not e-mail). You may post it anonymously if you wish. Often times, if one student has a question/comment, other also have a similar question/comment. Use e-mail with the professor, TA, graders only for issues that are specific to you individually (e.g., a scheduling issue or grade issue).

Catalogue Description: Fundamental concepts of algorithmic thinking as a primer to programming. Introduction to C++.

Course Description: This course introduces students to the fundamental concepts of algorithmic thinking as a primer to programming. It is intended for students who have little to no prior programming experience with the goal of providing a strong foundation for CS 103 Introduction to Programming. The course introduces the mathematics and basic language constructs needed for programming as well as the problem-solving techniques required to analyze a problem and produce an algorithm. These techniques are put into practice over the course of the semester with an introduction to programming using C++. Weekly lab and programming assignments will provide hands-on experience and active learning techniques.

Concepts include data representation, basic discrete math, control structures (conditional and iterative structures), functions, and arrays. Weekly small-group discussions will provide the opportunity for students to practice the concepts learned in class, review and ask questions. Weekly assignments will provide opportunity to practice, apply, and deepen the knowledge gained from lectures. By the end of this course, students should feel comfortable to take information-based problem descriptions and write a software program in C++ to perform the required task.

Course Objectives: Upon successful completion of this course a student will

- Choose appropriate data and variable types to store specific kinds and ranges of information.
- Write, compile, and run a computer program.
- Use basic discrete mathematics to understand, describe, and solve computation problems.
- Understand the way computers represent and operate on data.
- Trace provided C and C++ code line-by-line to analyze what operations are being performed and describe what the program will output.
- Employ programming concepts: variables, control structures, loops, and arrays to develop programs that solve information problems.

- Interpret written program requirements and develop a programmatic solution to meet those requirements.

Exam Dates:

- **Midterm Exam:** Thursday March 12, Section 1: 2:00-2:50 PM, Section 2: 3-3:50 PM.
- **Final Exam:** Section 1: Thursday, May 7, 2:00 PM- 4:00 PM
  Section 2: Tuesday May 12, 2:00 PM- 4:00 PM as set by the university.

Textbooks:

- **Required Textbook:**
  1. Frank Vahid and Roman Lysecky, *Programming in C++ with ZyLabs.* (Zy)
- **Recommended Textbook:**
  1. Cay Horstmann, *C++ for Everyone*, 2$^{\text{nd}}$ Edition, Wiley; 2012. (Hort)

Grading Policies:

- The letter grade distribution table guarantees the *minimum* grade each student will receive based on their final score. When appropriate, relative performance measures will be used to assign the final grade, at the discretion of the instructor.
  - Final grades are non-negotiable and are assigned at the discretion of the instructor. If you cannot accept this condition, you should not enroll in this course.
- Your three lowest grades in the Labs and two lowest scores in programming assignments will be dropped from the final grade. This policy gives a break to students who miss the deadline of a homework unintentionally or register late in the course or miss a homework because of poor time management. Missing any of the homework assignments is not a good idea if you want to learn the concepts.
- *Participation on Piazza has up to 5% extra credit, which is granted on a competitive basis *at the discretion of the instructor.*
- **Homework Policy**
  - Class Participation Activities will be assigned before each class and should be completed before class or shortly after class, depending on the due date.
  - Homework is assigned on an approximately weekly basis. A one-day grace period can be used for each homework with 10% penalty. *Absolutely no late homework will be accepted after the grace period. A late assignment results in a zero grade.*
  - In case of documented illness or grave family situations, exceptions can be made to the late submission policy.

  - Poor internet connection, failing to upload properly, or similar issues are **NOT** acceptable reasons for late submissions. If you want to make sure that you do not have such problems, submit homework *eight* hours earlier than the deadline. Please do not ask the instructor to make individual exceptions.
  - Students are encouraged to discuss homework problems with one another, but each student must do their own work and submit individual solutions written/ coded in their own hand. Copying the solutions or submitting identical homework sets is written evidence of cheating. The penalty ranges from F on the homework or exam, to an F in the course, to recommended expulsion.
  - Posting the homework assignments and their solutions to online forums or sharing them with other students is strictly prohibited and infringes the copyright of the instructor. Instances will be reported to USC officials as academic dishonesty for disciplinary action.

- **Exam Policy**
  - **Make-up Exams:** No make-up exams will be given. If you cannot make the above dates due to a class schedule conflict or personal matter, you must drop the class. In the case of a required business trip or a medical or family emergency, a signed letter from your manager or counselor or physician has to be submitted. This letter must include the contact of your physician or counselor or manager.
  - Midterm and final exams will be closed book and notes.
  - All exams are cumulative, with considerable emphasis on material presented since the last exam.

- **Attendance:**
  - Students are required to attend all the lectures and discussion sessions and actively participate in class discussions. Use of cellphones and laptops for anything unrelated to programming in C++ is prohibited in the classroom.

Important Notes:

- Textbooks are secondary to the lecture notes and homework assignments.
- Please use your USC email to register on Piazza and to contact the instructor and TAs.

Tentative Course Outline

Tuesday	Thursday
Jan 14th **1** **Introduction** (Zy 1.1-1.3) Programming (general introduction) Programming basics Comments and whitespace	16th **2** **Introduction** (Zy 1.4-1.7) Errors and warnings Computers and programs (general) Computer tour Language history
21st **3** **Introduction** (Zy 1.8-1.10) Problem solving Why programming? Why whitespace matters?	23rd **4** **Variables and Assignments** (Zy 2.1-2.5) Variables and assignments (general) Variables (int) Identifiers Arithmetic expressions (general) Arithmetic expressions (int)
28th **5** **Variables and Assignments** (Zy 2.7-2.11) Floating-point numbers (double) Scientific notation for floating-point literals Constant variables Using math functions Integer division and modulo	30th **6** **Variables and Assignments** (Zy 2.12-2.15) Type conversions Binary Characters Strings
Feb 4th **7** **Variables and Assignments** (Zy 2.16-2.18) Integer overflow Numeric data types Unsigned	6th **8** **Variables and Assignments** (Zy 2.19-2.22) Random numbers Debugging Auto Style guidelines
11th **9** **Conditional Statements** (Zy 3.1-3.4) If-else branches (general) If-else More if-else Equality and relational operator	13th **10** **Conditional Statements** (Zy 3.5-3.7) Detecting ranges (general) Detecting ranges with if-else statements Logical operators
18th **11** **Conditional Statements** (Zy 3.9-3.11) Order of evaluation Switch statements Boolean data type	20th **12** **Conditional Statements** (Zy 3.12-3.15) String comparisons String access operations Character operations More string operations

Tuesday	Thursday
25th **13** **Conditional Statements** (Zy 3.16-3.18) Conditional expressions Floating-point comparison Short circuit evaluation	27th **14** **Loops** (Zy 4.1-4.3) Loops (general) While loops More while examples
Mar 3rd **15** **Loops** (Zy 4.4-4.6) For loops More for loop examples Loops and strings	5th **16** **Loops** (Zy 4.7-4.9) Nested loops Developing programs incrementally Break and continue
10th **17** **Midterm Review***	12th **18** **Midterm**
17th Spring Recess	19th Spring Recess
24th **19** **Loops** (Zy 4.10-4.11) Variable name scope Enumerations	26th **20** **Arrays & Vectors** (Zy 5.1-5.5) Array/vector concept (general) Vectors Array/vector iteration drill Iterating through vectors Multiple vectors
31st **21** **Arrays & Vectors** (Zy 5.6-5.9) Vector resize Vector push-back Loop-modifying or copying/comparing vectors Swapping two variables (General)	Apr 2nd **22** **Arrays & Vectors** (5.11-5.15) Arrays vs. vectors Two-dimensional arrays Char arrays / C strings String library functions Char library functions: ctype
7th **23** **User-Defined Functions** (Zy 6.1-6.4) User-defined function basics Return Reasons for defining functions Functions with conditional statements/loops	9th **24** **User Defined Functions** (Zy 6.5-6.8) Unit testing (functions) How functions work Functions: Common errors Pass by reference
14th **25** **User-Defined Functions** (Zy 6.9-6.12) Functions with string/vector parameters Functions with C string parameters Scope of variable/function definitions Default parameter values	16th **26** **User-Defined Functions** (Zy 6.13-6.16) Function name overloading Parameter error checking Preprocessor and include Separate files

Tuesday	Thursday
21st **27** **Searching & Sorting Algorithms*** (Zy 15.1-15.4) Binary Search The Big O Notation Time Complexity	23rd **28** **Searching & Sorting Algorithms*** (Zy 15.5-15.9) Sorting Bubble Sort Selection Sort Quick Sort
28th **29** **Interpreted Languages*** Python	30th **30** **Review*** Final Tips

Notes:

- Items marked by * will be covered only if time permits.

Statement on Academic Integrity: USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles. SCampus, the Student Guidebook, contains the University Student Conduct Code (see University Governance, Section 11.00), while the recommended sanctions are located in Appendix A. See: `http://scampus.usc.edu`.

Emergency Preparedness/Course Continuity in a Crisis In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies. See the university's site on Campus Safety and Emergency Preparedness: `http://preparedness.usc.edu`

Statement for Students with Disabilities: Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. Website: `http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html`

(213) 740-0776 (Phone), (213) 740-6948 (TDD only), (213) 740-8216 (FAX) ability@usc.edu.