

**SABANCI UNIVERSITY**  
**CS 201 - Introduction to Computing**  
**Spring 2021-2022**

---

### **Instructors**

Gülşen Demiröz (Section A)

- FENS L015
- E-mail: [gulsen.demiroz@sabanciuniv.edu](mailto:gulsen.demiroz@sabanciuniv.edu)
- Lectures
  - Monday 09:40-11:30
  - Wednesday 12:40-13:30
  - <https://sabanciuniv.zoom.us/j/6290818441>

Barış Altop (Section B)

- FENS G001-C
- E-mail: [baris.altop@sabanciuniv.edu](mailto:baris.altop@sabanciuniv.edu)
- Lectures
  - Monday 15:40-17:30 (FENS G077)
  - Wednesday 16:40-17:30 (FENS G077)
  - <https://sabanciuniv.zoom.us/j/96970674346?pwd=NW9ZNnE0MEI5bVBiRFhxL2RNaWFFdz09>

### **IMPORTANT NOTE**

THIS SYLLABUS IS PREPARED FOR AN **HYBRID EDUCATION SEMESTER**. IN CASE OF A CHANGE OF EDUCATION METHODS (ONLINE, HYBRID, PHYSICAL, OR ANY OTHER MODEL) DURING THE SEMESTER DEPENDING ON NEW PANDEMIC RULES **THIS SYLLABUS WILL BE UPDATED/CHANGED ACCORDINGLY**.

### **Course Description**

The objective of this course is to introduce students to the field of computing and problem solving with the help of an object-oriented programming language (C++). Hence the course will cover many C++ features in detail as needed so students will also be learning a structured programming language.

Through the lectures, quizzes, take-home exams, and interactive recitations students will learn how to design algorithms based on object-oriented programming paradigms. Evaluation of the solutions in terms of correctness and efficiency will also be covered.

## Course Website

<https://sites.google.com/sabanciuniv.edu/cs201>

## TextBook

A Computer Science Tapestry, 2nd Edition, Owen L. Astrachan  
NOT available in the bookstore anymore but available at the library and online.  
We may not stick to the textbook all the time, you are responsible for all material covered in class.

## Course Tools and Installation Guide

- [Install Visual Studio 2012 on Windows](#)
- [Install Windows 10 & Visual Studio 2012 on macOS](#)
- [C++ Programming with Xcode on macOS](#)

## Tentative Grading (subject to change)

- Participation **7%**
- Take-home Exam **25%**
- Midterm Exam **33%**
- Final Exam **35%**

## Important Notice about grading:

### Weighted average is not the only criterion in letter grading!

- If your exams' *weighted* average ( $((MT * 0.33) + (FNL * 0.35))/0.68$ ) is below 30, you will fail the course even if your total grade is equal or above the overall passing grade. (Having exams' weighted average greater than or equal to 30 does not mean that you will pass the course.)
- There won't be any make-up for take-home exams in any case, including medical health reports and official university activities.
- If you miss all of the take-home exams, midterm and final exams as well; then you will get an NA grade.
- If you miss the midterm or the final exam and if you do not take the make-up exam for that missing exam; then you will directly get an F grade.

## Take-home Exams

There will be at least 5 or more take-home exams like mini programming projects. They will be assigned and collected at **SUCourse+**. All of the assigned take-home exams will be graded and taken into consideration in the overall grade. Recitations will be used for clarification about these.

Contribution of the take-home exams to the overall grade will be calculated according to the formula given below:

$$take\_home\_grade \begin{cases} take\_home\_averg & \text{if } ratio \leq 2 \\ take\_home\_averg \times (3 - ratio) & \text{if } 2 < ratio < 3 \\ 0 & \text{if } ratio \geq 3 \end{cases}$$

$$ratio = submitted\_take\_home\_avr / weighted\_exam\_averg$$

$$weighted\_exam\_averg = (0.33 \times midterm\_grade + 0.35 \times final\_grade) / 0.68$$

$$course\_numeric\_grade = participation \times 0.07 + take\_home\_grade \times 0.25 + \\ midterm\_grade \times 0.33 + final\_exam\_grade \times 0.35$$

Your programs will be automatically graded by GradeChecker (GC) or CodeRunner (SUCourse+), more details about GC is [here](#).

## Useful Information about course and FAQ

You can also find them [here](#).

## Expectations from Students

- Students are expected to attend all classes, recitations. Learning a programming language is much easier when a person explains it. You have to spend more time to compensate for a missed class. So absenteeism does not buy time. You must attend the hybrid lectures, online recitations and physical exams.
- Students are responsible for the material covered in class even if it is not part of the lecture notes published on the website (Sometimes we explain some stuff on the board). That is why attendance is important.
- Students are responsible to check their emails (Sabanci University accounts), SUCourse and course website daily for any announcements related to this course.
- **PLAGIARISM WILL NOT BE TOLERATED.**
  - Any act of plagiarism may result in a direct fail (F) of the course.
- If you need to take a medical report for any exam; Please read the policies on both **plagiarism** and **make-up exams** [here](#).

## Course Outline

- Week 1: Introduction to Programming Languages, Data Representation (bits and bytes), basic programming structure and concepts: identifiers, literals, symbols, variables, screen input/output (cin and cout)
- Week 2: Basic data types (int/double/char/bool) and basic arithmetic operations with their precedence, first C++ program with Visual MS Studio
- Week 3: Functions with/out return values, function prototypes, parameter passing (pass by value and by reference)
- Week 4: Conditional statements (if-else), nested else-if statements, logical operators (&&, ||, !)
- Week 5: String class, loops (while, for, do-while)
- Week 6: Loops (while, for, do-while), char data type
- Week 7: File I/O, console stream cin, input and output file streams, string streams
- Week 8: *Midterm (Tentative, Lectures may continue)*
- Week 9: Structs, enum, vectors/arrays and vector operations: sequential/binary search, insert/delete to a vector
- Week 10: Sorting (selection and insertion sort) and introduction to algorithm complexity analysis, vector of structs, matrix
- Week 11: Classes and objects: using existing classes such as Dice, RandGen, Date
- Week 12: Classes and objects: using and modifying existing classes such as Dice, RandGen, Date
- Week 13: Recursion
- Week 14: Pointers, linked lists