**CS 170 Language C++ Section 71**

**Course Syllabus, Spring Semester 2022**

**Time and Venue:** TF 5:30 to 7:10 in room HI 031

**Credits:** 4 required CS credits

**Prerequisites:** CS 140 with a grade of (C) or better

**Instructor:** Dr. Gancho Ganchev, Office: HI 039G

TF 3:15 – 5:30 pm, W 6:30-7:00 pm

WebEx: wcsu.webex.com/meet/ganchevg

Phone: (203) 837-9349

Email: [ganchevg@wcsu.edu](mailto:ganchevg@wcsu.edu)

**Description:**

An in-depth introduction to a modern programming language (C++) and its use in constructing programs that solve a variety of problems. Students entering the course should be familiar with fundamental programming techniques in another language, including using variables and data types, branching, looping, and one-dimensional data collections. These topics will be reviewed and extended with regard to the language C++. Other topics covered in the course include streams, files, pointers, dynamic memory allocation, scoping, argument passing mechanisms, functions as arguments, function and operator overloading, template functions, recursion, classes and other user-defined data types, vectors, multidimensional arrays, strings and C-strings. The software development process is emphasized. The course will include a significant amount of project work with relatively short deadlines. Please plan sufficient time for regular independent work, at least twice as much as the time for scheduled class sessions per week, every week.

**Resources:**

Required Textbook:

[1] *Big C++* by Cay Horstmann and Timothy Budd, Wiley 2017, ISBN 978-1-119-40297-8

* In this course we will use the first 11 chapters of the book. The remaining chapters will be used in CS 221.

Class notes and additional course materials will be posted on the N: drive in the folder N:/Class/GanchevG/CS170

Additional Reading / Exercises:

[2] *Absolute C++* by Walter Savitch et al, Pearson 2016

[3] *C++ How to Program* by H. M. Deitel and P. J. Deitel, Pearson 2017.

[4] *Introduction to Programming with C++* by Daniel Liang, Pearson 2014

**Learning outcomes:**

* Develop non-trivial C++ programs
  + Master the foundations of a widely used programming language, C++
  + Compare and contrast the object-oriented and the procedural approach of software development
  + Create iterative and recursive algorithms for solving non-trivial problems (e.g., searching, sorting, reversal).
  + Efficiently use the features of the C++ programming language to implement, test, and debug algorithms for solving non-trivial problems.
* Apply basic principles of good human-computer interaction
* Test and debug programs of moderate complexity
  + Apply a variety of strategies to the testing and debugging.
  + Identify common coding errors such as buffer overflows and memory leaks and apply strategies for avoiding such errors.
  + Create unit test plans for program modules.
* Apply consistent documentation and program style standards that contribute to the readability and maintainability of software

**Attendance:** You are expected to be present for each class session.

**Withdrawals:** No withdrawals after the official WCSU deadline.

**Assessment:** Two tests and three programming assignments, each consisting of three to four independent small programming projects. The emphasis will be on producing high quality programs with good internal and external documentation and efficient utilization of language features. No make-up tests will be offered during the semester. If you miss a test for a valid reason, a make-up test will be offered during the final exams’ week. Assignments will be due about 2 weeks after posting.

NOTE: Homework exercises are assigned at nearly every session. You are strongly encouraged to perform the assigned work and submit it for feedback. There will be one required homework submission in the first half of the semester. There will be a review test in the beginning of the semester. These two items will not count towards your final grade.

**Grading:** Assignment 1. . . . . . . . . . 15%

Test 1 . . . . . . . . . . . . . . . . 15%

Assignment 2 . . . . . . . . . . 20%

Assignment 3 . . . . . . . . . . 25%

Test 2 . . . . . . . . . . . . . . . . 25%

**Tentative Schedule of Topics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Session | Date | Topics | Reference | Note |
| 1-Fr | Jan 21 | Introductions. Introduction to the language and the development environment. | Notes, [1] Ch. 1, 2 |  |
| 2-Tu | Jan 25 | Primitive data types, expressions, assignment. | Notes, [1] Ch. 2 |  |
| 3-Fr | Jan 28 | User-defined data types: enums.Simple functions. | Notes, [1] Ch. 2 |  |
| 4-Tu | Feb1 | Review TestAddress operator, pointers. Simple dynamic memory management. Using strings and other objects. | Notes, [1] Ch. 2 |  |
| 5-Fr | Feb 4 | Input and output with streams | Notes |  |
| 6-Tu | Feb 8 | C++ control statements. Input validation. Stream error flags and functions. Exercises. | Notes, [1] Ch. 3 |  |
| 7-Fr | Feb 11 | C++ control statements. The ? conditional operator. Exercises. | Notes, [1] Ch. 3 |  |
| 8-Tu | Feb 15 | Control Statements. Generating random numbers. |  |  |
| Fr | Feb 18 | **President’s Holiday – no class** |  |  |
| 9-Tu | Feb 22 | Control Statements. Exercises. | Notes, [1] Ch. 3 |  |
| 10-Fr | Feb 25 | **Submit homework for feedback**  Control Statements. Exercises  **Assignment 1 out** | Notes, [1] Ch. 3 |  |
| 11-Tu | Mar 1 | Testing and documentation.  Functions. Passing arguments by value. | Notes, [1] Ch. 4 |  |
| 12-Fr | Mar 4 | Scope and lifetime of variables. | Notes, [1] Ch. 4 |  |
| 13-Tu | Mar 8 | Passing arguments by reference.  Function pointers. | Notes, [1] Ch. 4 [1] Notes, Ch. 7 |  |
| 14-Fr | Mar 11 | **Assignment 1 due by 5:00 pm**  Recursive functions. Exercises. | Notes, [1] Ch. 10 |  |
|  | Mar 14-20 | Spring Break  Midterm grades due Mo March 21 |  |  |
| 15-Tu | Mar 22 | More functions: overloading, templates, variable number of arguments, default arguments, functions vs. macros. Assertions. | Notes |  |
| 16-Fr | Mar 25 | More functions.Exercises |  |  |
| 17-Tu | Mar 29 | Test 1 |  |  |
| 18-Fr | April 1 | One-dimensiona arrays. Selection sort. Bubble sort.  Assignment 2 out | Notes, [1] Ch. 6,7 |  |
| 19-Tu | Apr 5 | Merge sort, qsort. One-dimensional arrays and pointers. Exercises. |  |  |
| 20-Fr | Apr 8 | Multi-dimensional arrays. More dynamic memory allocation. Heap-dynamic 2-dimensional arrays. Exercises. | Notes, [1] TBA |  |
| 21-Tu | Apr 12 | Vectors. Exercises. |  |  |
| 22-Fr | Apr 15 | University closed |  |  |
| Mo | Apr 18 | Last date for withdrawal (but please consult me before deciding to withdraw!) |  |  |
| 23-Tu | Apr 19 | Assignment 2 due by 5:00 pm  C-Strings and library functions. Arrays of c-strings, use of pointers. Command-line arguments.  Assignment 3 Part 1 out | Notes, [1] Ch 5 |  |
| 24-Fr | Apr 22 | Classes. Exercises. | Notes, [1] Ch 9 |  |
| 25-Tu | Apr 26 | Files: work with text and binary files, sequential and random access. Structs | Notes, [1] Ch 5 |  |
| 26-Fr | Apr 29 | Files: work with text and binary files, sequential and random access. Structs.  Assignment 3 Part 2 out | Notes, [1] Ch 5 |  |
| 27-Tu | May 3 | Overloading commonly used operators in user defined types (examples with enumerations and classes). Exercises. | Notes, [1], Ch 7 |  |
| 28-Fr | May 6 | Object Oriented Programming: Introduction to Inheritance | Notes,  [1] Ch 5, 8 |  |
| Tu | May 10 | Assignment 3 due by 5:00 pm |  |  |
| Fr | May 13 | Test 2 (5:30-8:00 pm) |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

NOTE: The above schedule may be adjusted during the semester to reflect the time actually spent with topics.

**Disability Accommodation Statement**

*If you have a disability and would like to request accommodations, please visit AccessAbility Services, located in Higgins Annex 017. They will give you an accommodation letter which you should bring to me as soon as possible.*