C/C++ Programming I Syllabus

Course Information

Course Number: CSE-40475
Section ID: 146359
Quarter: Spring 2020
Course Dates: 3/31/2020 – 5/30/2020
Last Date to Drop with a Refund: 4/6/2020
Credit units: 3.0

Instructor Information

Name: Ray Mitchell
E-mail: MeanOldTeacher@MeanOldTeacher.com

Instructor/Student Communication

Instructor/student communication is via e-mail only and is strongly encouraged. There are no specific “office hours” but inquiries are answered as quickly as possible, often within minutes.

Course Description and Objectives

This course provides a working knowledge of data types, basic operations, standard library functions, portability issues, standard programming practices, and style. C and C++ are covered concurrently with C++ being presented as a "safer" version of C that offers an extended set of features and capabilities rather than as an object-oriented language. Object-oriented programming is taught in the **C/C++ Programming III** and **IV** courses.

Course Prerequisites

Recent completion of an introductory programming course in any language or equivalent experience is recommended. Students are expected to be familiar with general concepts such as logical problem solving, constants, variables, arithmetic operations, loops, and functions.

Course Materials

Operating Systems and Program Development Tools:
This course is not tied to any specific operating system or program development tools and students may choose according to their own needs and preferences. The most commonly chosen free products are "Visual Studio Community" for Windows, "Xcode" for macOS, and "Code::Blocks" for Linux, Windows, and macOS.

Drawings, Diagrams, and PDF File Conversion:
Some assignments require drawings/diagrams in PDF format that must either be machine-produced or neatly drawn by hand and digitized. PDF converters are readily available as well as being built into many applications such as Microsoft Word, Visio, etc.

Required Book:
The only book required for this course is titled "Beginning & Advanced C/C++ Notes" and students must refer to it as they listen to the course's audio lessons. It is only available from the UCSD Bookstore and includes a hard copy and an online version. For convenience students are initially provided with the sections necessary for the first three lessons. Since this book is only a collection of notes pertaining to the course lessons and homework and is not intended as a comprehensive textbook, supplemental textbook recommendations are provided for students wanting more comprehensive information.

E-mail Is Required:
An e-mail system capable of receiving TXT, PDF, and ZIP file attachments is required. Systems that strip or corrupt any of these types are not acceptable. Students must always check their spam mailboxes if an expected e-mail is not received!

Assignment Information

- There are 8 assignments and their content may differ between students to protect course integrity. All are provided via e-mail at the beginning of the course.
- Assignments are due by the deadlines indicated below but may be submitted early.
- Late submissions for any reason other than a verifiable medical/family emergency, U.S. military obligation, or late enrollment will receive a significant credit deduction. Please do not request exemptions for any other reason.
- Submissions are ONLY accepted for grading via the "assignment checker" and will not be graded or receive credit otherwise. Submission details will be provided.
- Solutions will be included with each graded assignment I return.

Assignment Submission Deadlines			
Asg #	Related Notes	All Are Tuesdays @ 11:59PM	Time Zone
1	1.1 – 1.18	April 7	PDT
2	2.1 – 3.7	April 14	PDT
3	3.8 – 4.3, D.1 – D.9	April 21	PDT
4	5.1 – 5.19	April 28	PDT
5	6.1 – 6.13	May 5	PDT
6	6.14 – 7.7	May 12	PDT
7	7.8 – 9.11	May 19	PDT
8	9.12 – 10.8	May 26	PDT
9	10.9 – 10.15	*no assignment 9*	--

Grading Policies

CHEATING WILL NOT BE TOLERATED IN ANY FORM!
Please see the next page for more information.

Late Submission Deductions:
You may resubmit exercises to the assignment checker as many times as you wish before the deadline. However, be wary of resubmitting late just to correct minor issues since I only grade the newest submission of each exercise even if there were previous on-time submissions. Late deductions are substantial, non-refundable, and are as follows:

Late Submission Deductions	
Hours Late	Deduction
< 3	50%
3-24	80%
> 24	100%

NOTE:
Your choice of any one, and only one, of assignments 1-7 (but not 8) may be submitted up to five days late without penalty. However, I must be notified prior to that assignment's original deadline.

Refundable Deductions:
For assignments 1 and 2 only, exercise deductions followed by the letter '**s**' will be refunded if all causes are completely corrected and the exercise is resubmitted to the assignment checker prior to the next assignment's deadline. No other deductions will be refunded for any exercise unless I have made an error.

Course Grading Scale:
Each of the 8 assignments is worth 20 points and the final course grade is based entirely upon the sum of the points earned on them as indicated in the table below. There are no exams, dropped scores, extra credit, or makeup work.

Final Course Grades			
Points	Percent	Course Grade	Status
>= 152	95%	A+	pass
>= 136	85%	A	pass
>= 128	80%	B+	pass
>= 104	65%	B	pass
>= 96	60%	C+	pass
>= 80	50%	C	pass
>= 77	48%	C-	pass
< 77	48%	F	fail

CHEATING WILL NOT BE TOLERATED IN ANY FORM!
ASSIGNMENTS ARE 1-PERSON PROJECTS –
NOT FRIEND / PARTNER / GROUP / TEAM EFFORTS

What cheating is:

working on a solution with someone else
–or–
submitting a solution you did not independently develop
–or–
sharing a solution with another student

What cheating is not:

occasionally discussing general strategies and
coding issues with others but doing entirely your own work
–or–
getting instructor help

CHEATING ON ANY EXERCISE WILL RESULT IN A SCORE OF 0 FOR THE ENTIRE ASSIGNMENT AND POSSIBLY THE ENTIRE COURSE AS WELL AS NOTIFICATION OF THE UCSD EXTENSION REGISTRAR.
I RESERVE THE RIGHT TO REGRADE PREVIOUS ASSIGNMENTS IF CHEATING IS DETECTED IN A LATER ASSIGNMENT.

Course Topical Outline

The following topics are scheduled to be covered during the lessons indicated but are subject to change based upon class needs. Note numbers refer to notes in the required course book.

1. *Notes 1.1 – 1.18*
   Introduction and Fundamentals
   Character and String Literals
   Compound Assignment
   Elements of a Simple Program
   Console I/O

2. *Notes 2.1 – 3.7*
   Arithmetic Types & Conversions
   The **sizeof** Unary Operator
   #include and #define
   const-qualified Variables
   Boolean Type
   Relational and Logical Statements
   for Statements

3. *Notes 3.8 – 4.3, D.1 – D.9*
   while and **do** Statements
   break and **continue** Statements
   if and **if/else** Statements
   switch Statements
   Character Analysis Functions
   Flags
   I/O Redirection
   End of File and EOF
   Preprocessor Topics

4. *Notes 5.1 – 5.19*
   Function Syntax
   Pass/Return Mechanism
   Function Prototypes
   Default Arguments
   Function Overloading
   Call By Value
   Reference Variables
   Automatic and **static** Variables
   External Variables
   Communication between Files
   static Externals and Functions
   Function-like Macros using #define
   inline Functions

5. *Notes 6.1 – 6.13*
   The Right-Left Rule
   One-dimensional Arrays
   Pointer Basics
   Pointers and Functions
   Call/Return by Reference
   const-qualified Call/Return

6. *Notes 6.14 – 7.7*
   Arithmetic Operations on Pointers
   Pointers and Arrays
   Functions and Arrays
   Strings and String I/O
   Library String Functions
   The **string** class

7. *Notes 7.8 – 9.11*
   Multidimensional Arrays
   Ragged Arrays
   Command Line Arguments
   Dynamic Storage
   Dynamic Ragged Arrays
   The **typedef** Facility
   Enumerated Data
   Structures

8. *Notes 9.12 – 10.8*
   Structures, cont'd.
   Data & Function class Members
   Text and Binary Files
   File Handling Functions
   Operations on Text Files

9. *Notes 10.9 – 10.15B*
   Operations on Binary Files
   File Descriptor I/O