SAN DIEGO MESA COLLEGE

CISC 192 C++ PROGRAMMING - FALL 2012 - CRN 89577 - 4 UNITS

MW: 2:20 PM-3:40 PM (Lecture, K201A) + 3:50 PM-5:15 PM (Lab, K406)
Class meets: 8/20/2012 to 12/17/2012

COURSE SYLLABUS

Programming is best learned by actually doing it

INSTRUCTOR:

Viet Quoc Trinh

Master of Science in Software Engineering,
Bachelor of Science in Computer Science

Phone: (858) 695-2693
Voice: (619) 388-2767 x 5203
Office Hour: MW: 12:00 PM-2:00 PM – Room: I3-412A
Mailbox: K108B - San Diego Mesa College, 7250 Mesa College Dr., San Diego, CA 92111
Website: http://classroom.sdmesa.edu/vtrinh/
Email:vtrinh@sdccd.edu

COURSE DESCRIPTION:

This course presents basic programming concepts using the C++ programming language. The organization of standard I/O classes is emphasized. Structured and Object oriented programming techniques are presented and used to design and implement a variety of programming problems. (FT). Associate Degree Credit & transfer to CSU and/or private colleges and universities.
UC Transfer Course List. CAN CSCI 16 = CISC 192

COURSE OBJECTIVES:

Upon successful completion of the course, students will be able to:

1. Define data types, constants, and arithmetic operations.
2. Use variables and objects including declaration and instantiation, data assignment, and the use of cin and cout.
3. Use relational and logical operators and create Boolean expressions.
4. Use decision structures including if-else and switch.
5. Apply iteration structures including while, for, and do-while.
6. Create user-defined functions including parameter lists and returning values.
7. Create arrays, data assignment, data manipulation, and output from arrays.
8. Use arrays as parameters in user-defined functions.
9. Employ searching and sorting arrays.
10. Create and manipulate strings including character access and string parsing.

11. Use standard string manipulation functions including strcmp(), strcpy(), and strlen().
12. Employ pointer variables and describe the relationship between pointers, standard variables and arrays.
13. Create structures, arrays of structures, and a signally linked list of structures.
14. Use data file including writing and reading to and from both text and binary files.

STUDENT LEARNING OUTCOMES:

Upon successful completion of the course, students will be able to:

1. Understand from a C++ perspective the control of program flow based on the conditional evaluation of a boolean expression.
2. Use C++ to provide an if… else structure as a solution to a software problem.
3. Understand from a C++ perspective the control of program flow with a looping structure.
4. Use C++ to provide a looping structure as a solution to a software problem.

REQUIRED TEXT & SUPPLIES:

- ✓ Tony Gaddis. **Starting Out with C++: From Control Structure through Objects.** 7th Edition. ISBN-10: 0132576252 (*ISBN-13: 9780132576253*). Publisher: Addison-Wesley. 2012
  Book's website: http://www.pearsonhighered.com/educator/product/Starting-Out-with-C-From-Control-Structures-through-Objects/9780132576253.page
- ✓ 3 Scantrons
- ✓ Microsoft Visual C++ 2010 Express
- ✓ 1 USB drive

METHOD OF INSTRUCTION:

- ✓ Computer assisted instruction
- ✓ Lecture
- ✓ Audio/Visual aided instruction
- ✓ In-class computer hands-on practice of concepts and techniques included in course objectives
- ✓ Interactive group activities including analysis, evaluation, and modification of C++ programs

METHOD OF EVALUATION:

A student's grade will be based on multiple measures of performance and will reflect the objectives set forth in this outline. A final grade of "C" or better indicates the student has the ability to successfully apply theory and techniques taught in this course in subsequent courses and in practice. Evaluation methods may include, but are not limited to the following:

- ✓ Performance on hands-on assignments
- ✓ Written responses to in-class assignments

- ✓ Responses to in-class objective and/or essay question quizzes and/or Examinations
- ✓ Development of programs in C++ programming language
- ✓ Interactive one-on-one demonstration of program testing and operations
- ✓ Participation in classroom discussion
- ✓ Development of program documentation

WEIGHTING OF ASSIGNMENTS:

- ✓ 3 Exams (150 points - 50/each)
- ✓ 14 C++ Programming Projects (150 points)

PP1	PP2	PP3	PP4	PP5	PP6	PP7	PP8	PP9	PP10	PP11	PP12	PP13	PP14
5	10	10	10	10	10	10	10	10	10	15	15	15	10

Total Points: 300

GRADING SCALE:

A = 270 to 300 (>= 90%)	**B** = 240 to 269 (>= 80%)	**C** = 210 to 239 (>= 70%)
D = 180 to 209 (>= 60%)	**F** = 179 or below (< 60%)	*Grades are not given. They are earned.*

CLASS-SPECIFIC STANDARDS:

- ✓ Each assignment will be due on or before the next assignment's assigned date, at the lab time, or email to me at vtrinh@sdccd.edu
  Late assignments receive a 10% deduction for each day they are late. Assignments more than 5 days late will not be accepted.
- ✓ All C++ programs/ projects must be commented well - 2 points off for without or just a few comments.
- ✓ Class begins at 2:20 PM. In a professional setting, tardiness is not acceptable. Tardiness is also disruptive to the learning environment.
  Promptness reflects professional courtesy and is expected.
- ✓ Class attendance is required. It is the student's responsibility to add, drop or withdraw from classes before the deadlines stated in the class schedule. It is the student's responsibility to drop all classes in which he/she is no longer attending.
  If you miss class, make arrangements with a classmate to keep you informed on lecture topics and lab exercises.
  The final grade in this class will be affected by active participation, including attendance, as

follows:
Excessive absences, more than 10 times/ 5 weeks (*accumulated*), will be reduced to lower one letter grade.
Choose ONLINE class instead, if you think you could have many absences.
It is the instructor's discretion to withdraw a student after the add/drop deadline (*August 31, 2012*) due to excessive absences.

- ✓ Students who remain enrolled in a class beyond the published withdrawal deadline, as stated in the class schedule (*October 26, 2012*), will receive an evaluative letter grade in this class.
- ✓ Any special circumstance will be considered, however it must be arranged in advance, or informed me immediately if it happened; proof will be asked to verify.
- ✓ Any discriminations regarding to gender, religion, physical challenge, and race are not tolerable in any levels.
- ✓ Read the textbook and take notes. Read the related sections before coming to class. Your notes are allowed to use in the exams.
  Prepare, bring and discuss your possible questions in lectures.
  Be courteous; one person can talk at a time. You will be asked to leave for your disruption.
- ✓ Lab time is mandatory. All programming assignments will be helped, collected and graded at the lab time.
- ✓ ***Important Deadline Dates:***
  - *Add*: August 31, 2012
  - *Drop with Refund*: September 04, 2012
  - *Drop without "W"*: August 31, 2012
  - *Withdrawal*: October 26, 2012
- ✓ All cell phones should be set to silent/vibrate mode in the lecture time.
- ✓ ***Cheating/Plagiarism:***
  Students are expected to be honest and ethical at all times in the pursuit of academic goals. Students who are found to be in violation of Administrative Procedure 3100.3 Honest Academic Conduct, will receive a grade of zero on the assignment, quiz, or exam in question and may be referred for disciplinary action in accordance with Administrative Procedure 3100.2, Student Disciplinary Procedures.
- ✓ ***Students with Disabilities:***
  All students with disabilities requiring accommodations are responsible for making arrangements in a timely manner through the Center for Students with Disabilities.

TIME TABLE:

Week	Date	Subjects/ Topics - Lectures	Assignments - Labs
1	8/20	Syllabus - Add Code - Chapters 1 & 2: Introduction to C++ Programming	Obtain Student Account

	8/22	Chapter 3: Expressions and Interactivity	PP #1: Expression
2	8/27	Chapter 4: Making Decisions	PP #2: Selection Structure
	8/29	Chapter 5: Loops and Files	PP #3: Repetition Structure
3	9/3	**LABOR DAY - CLASSES NOT IN SESSION**	
	9/5	Chapter 6: Functions	PP #4: Procedural Structure
4	9/10	Chapter 6: (continued)	
	9/12	Chapter 7: Arrays	PP #5: Array
5	9/17	Chapter 8: Searching and Sorting Arrays - Exam 1 Review	
	9/19	**Exam 1 (Chapters 1-7)**	
6	9/24	Chapter 9: Pointers	PP #6: Searching/Sorting Array
	9/26	Chapter 9: (continued) -	
7	10/01	Chapter 10: Characters, C-strings and string class	PP #7: Pointer
	10/03	Chapter 10: (continued) -	
8	10/08	Chapter 11: Structured Data -	PP #8: String
	10/10	Chapter 12: Advanced File Operations -	
9	10/15	Chapter 13: Introduction to Classes	PP #9: File
	10/17	Chapter 13: (continued)	
10	10/22	Chapter 13: (continued) -	PP #10: Abstract Data Type
	10/24	**Exam 2 (chapters 8-12)**	

11	10/29	Chapter 14: More About Classes	
	10/31	Chapter 14: (continued)	
12	11/05	Chapter 15: Inheritance	PP #11: Object Oriented Programming
	11/07	Chapter 15: Polymorphism	
13	11/12	**VETERAN - CLASSES NOT IN SESSION**	
	11/14	Chapter 16: Exceptions, Templates, and the Standard Template Library	PP #12: Inheritance/Polymorphism
	11/19	**THANKSGIVING - CLASSES NOT IN SESSION**	
	11/21	**THANKSGIVING - CLASSES NOT IN SESSION**	
14	11/26	Chapter 16: (continued)	
	11/28	Chapter 17: Linked Lists	PP #13: Linked List
15	12/03	Chapter 17: (continued)	
	12/05	Chapter 19: Recursion	PP #14: Recursion
16	12/10	Exam 3 Review	
	12/12	**Exam 3 (Chapters 13-17 & 19)**	All Assignments must be turned in

ADDENDUN FOR HONOR PROGRAM:

Honor students must have three extra programming assignments that require the ability to research for related concepts and materials to implement the challenging tasks. The assignments must be done with object-oriented-programming style and deal with real things.