



ALAMO COLLEGES DISTRICT
Palo Alto College

PROGRAMMING FUNDAMENTALS II COSC-

1437

Full Term Spring 2016 Section 004.25830 4-4-0 Credits 01/19/2016 to 05/14/2016

Modified 12/18/2015

MEETING TIMES

This course meets on-line.

CONTACT INFORMATION

INSTRUCTOR : Dr. Neven Jurkovic
OFFICE : Sabine 205
OFFICE HOURS : Th: 5:00pm - 7:00pm
PHONE : 486-3316
E-mail : njurkovic@alamo.edu

*(Current students : For
faster response, please
contact the instructor via*

MATERIALS

Starting out with C++: From Control Structures through Objects (7th edition; Brief Edition) by Tony Gaddis (isbn: 978-0132772891, pub.: Pearson)

DESCRIPTION

Review of control structures and data types with emphasis on structured data types. Applies the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Includes basic analysis of algorithms, searching and sorting techniques, and an introduction to software engineering. (Formerly known as COSC 2430/2318 Advanced Structured Programming Techniques.) Students cannot receive credit for both COSC 1337 and COSC 1437.

Prerequisites

COSC 1315 or COSC 1336

OBJECTIVES

The primary goal of this course is to develop in the student an understanding of programming logic and advanced programming techniques. A programming language will be used as the medium in which the student will express these techniques. A second goal, therefore is for the student to learn a programming language at the advanced level and be able to express his/her programming logic in that language.

OUTCOMES

- 1 Identify and explain a programming development lifecycle, including planning, analysis, design, development, and maintenance.
- 2 Demonstrate a basic understanding of object-oriented programming by using structs and classes in software projects.
- 3 Use object-oriented programming techniques to develop executable programs that include elements such as inheritance and polymorphism.
- 4 Document and format code in a consistent manner.
- 5 Apply basic searching and sorting algorithms in software design.
- 6 Apply single- and multi-dimensional arrays in software.
- 7 Use a symbolic debugger to find and fix runtime and logical errors in software.
- 8 Demonstrate a basic understanding of programming methodologies, including object-oriented, structured, and procedural programming.
- 9 Describe the phases of program translation from source code to executable code.

EVALUATION

The course grade is based on the following:

Programming assignments (equally weighted)	50%
Midterm	30%
Final	20%
Total	100%

In addition to the programming assignments, BioSig identification has to be performed several times during the semester.

You need to complete your first BioSig identification on the first day of class. This is very important. Students who miss this deadline, will be considered absent and dropped as per College policy.

You also need to turn in your first assignment on time. Students who do not, will be dropped for non - participation.

Grading scale used:

90% - 100%	A
80% - 90%	B
70% - 70.9%	C
60% - 69.9%	D

COURSE POLICIES

- Students who do not turn in their first assignment on time will be dropped for non-participation
- Late assignments are not accepted, except in case of a documented medical emergency
- Missed exams can not be made up, except in case of a documented medical emergency

SCHEDULE

Module-by-Module Activities Schedule for C+ + Students

Misc. Comments on assignments and other activities:

- Recommended (optional) end-of-chapter activities include answering multiple choice review questions, and "what will this code do" type of questions. This should help you with your exams.
- Check the exact assignment and exam due dates on the Canvas calendar

- Detailed assignment submission guidelines are given [here](#).
- [This document](#) briefly describes what kind of help you can expect with your assignments
- If you are using an older edition of the book, to avoid the pages discrepancies, use this [assignment document](#).
- Exams can not be made up. Please read the [make-up policy here](#).

Module list has been truncated. See complete list at end of this document.

Module Start date	Textbook Chapter	Assignments and Exams	Assignments - additional information
6/5	1.6 - 1.7 [Intro - programming process, OOP, software engineering] 7 [arrays]	Download and install C++ compiler. Click for instructions Setting up your compiler (non credit assignment)	You need to complete your first BioSig identification on the first day of class. This is very important. Students who miss this deadline, will be considered absent and dropped as per College policy. You also need to turn in your first assignment on time. Students who do not, will be dropped for non-participation.
6/6	8 [sorting and searching arrays, algorithm efficiency]	A1 - Arrays "Sorting Benchmarks", pg. 488	Use the following function prototypes: int bubbleSort(long [], int); int selectionSort(long [], int); <i>Note: int in argument list is meant for array size. Function type is int because the function returns number of exchanges</i> Turn in the following files: a1main.cpp program compile and run screenshots design document
	9.1 - 9.6 [pointers, pointer arithmetic.		

b/ /	initializing, comparing]		
6/8	9.7 - 9.10 [pointers as function parameters, dynamic memory allocation, returning pointers]	A2 - Pointers "Movie Statistics", pg. 539 (see previous problems instructions (8,9) in regards to median and mode functions)	Use the following function prototypes: double median(int *, int); int mode(int *, int); int *makeArray(int); void getMovieData(int *, int); void selectionSort(int [], int); double average(int *, int); Turn in the

INSTITUTIONAL POLICIES

STUDENT RESPONSIBILITIES:

A. Attendance:

Effective Spring Term 2010, student absences will be recorded from the first day the class meets. Regular and punctual attendance in all classes and laboratories, day and evening, is required. Students who are absent for any reason should always consult with their instructors. Course syllabi must provide specific information regarding attendance, including, for courses involving the internet, online activity that constitutes "attendance." Also, both tardiness and early departure from class may be considered forms of absenteeism. In all cases, students will be held responsible for completion of course requirements covered in their absence.

Additionally, it is the student's responsibility to drop a course for nonattendance. Course instructors establish policy with regard to attendance in their respective syllabi and may drop a student for excessive absences. Absences are considered excessive when more than 12.5 percent of the total contact hours of instruction in a semester, including lecture and lab, are missed. For example, in a three-credit-hour lecture class, students may be dropped after more than six contact hours of absences. In a four-credit-hour lecture/lab class, students may be dropped after more than eight contact hours of absences. Absences are counted regardless of whether they occur consecutively.

In special programs with additional accreditation or certification standards, additional attendance requirements may be enforced but faculty must clearly explain these policies in their syllabi. Students who stop attending class for any reason should contact the instructor and the college registrar to officially withdraw from the class. Students may be required to consult with an advisor or designee before dropping.

Failure to officially withdraw may result in a failing grade for the course. It is the student's responsibility to withdraw officially from a class by submitting a completed Withdrawal Form to the Admissions and Records Office.

B. Student Responsibility for Success (Alamo Colleges Policy F.6.2):

As members of the Alamo Colleges learning community, students, faculty, staff and administrators all share the responsibility to create an atmosphere where knowledge, integrity, truth, and academic honesty are valued and expected. A clear acknowledgment of the mutual obligations of all members of the academic community emphasizes this implicit partnership in fostering the conditions necessary for student success.

In this relationship, the Alamo Colleges provides institutional policies, procedures, and opportunities to facilitate student learning that encourage interaction, involvement and responsible participation. Inherent in the academic climate is the expectation that students will assume responsibility for contributing to their own development and learning. Academic success is directly tied to the effort students put into their studies, the degree to which they interact with faculty and peers, and the extent to which students integrate into the campus life.

1. Engagement

1. Create connections and build relationships with faculty, staff and students (visit during office hours, join clubs and organizations, participate in student activities, etc.);
2. Stay informed of policies, procedures, deadlines and events for academic and co-curricular activities;
3. Complete all requirements for admission, registration, and payment by deadlines;
4. Apply for financial assistance, if needed, complying with all federal, state and local regulations and procedures;
5. Meet all federal, state and local health care regulations.

2. Communication

1. Seek guidance from faculty, advisors or counselors for questions and concerns in regards to degree plans, major selection, academic status, grades, and issues impacting college success;
2. Develop a peer support system to identify student contacts for questions, group assignments, etc. regarding academic and co-curricular activities;
3. Communicate with College personnel promptly regarding academic or co-curricular concerns and assistance requests;
4. Carefully consider the information provided by College personnel and make decisions using that information;
5. Check the Alamo Colleges' Web Services regularly for emails, holds, student records, financial aid status and announcements;
6. Submit disability documentation if seeking services and request academic accommodations in advance of each semester.

3. Academic Success

1. Complete courses with passing grades and maintain in good academic standing (2.0 GPA) status;
2. Read and follow all syllabi;
3. Purchase textbooks and required supplies in a timely manner;
4. Attend classes regularly and on time, with as few absences, late arrivals, and early exits as possible;
5. Arrive to class with all needed materials and completed assignments for that class period;
6. Be attentive in class and actively participate as appropriate;
7. Devote sufficient time for studying;
8. Ensure integrity in all aspects of academic and career development;
9. Accurately represent one's own work and that of others used in creating academic assignments. Use information ethically and exercise appropriate caution to avoid plagiarism on all assignments;
10. Notify faculty in advance or as soon as possible about absences and provide documentation as appropriate;
11. Consult faculty members in advance when unable to complete projects, assignments, or take examinations as scheduled.

4. Self-Responsibility and Responsibility to Others

1. Maintain accurate and complete degree/certificate major selection and contact information including name, address, phone number and emergency contact;

2. Balance personal obligations and educational pursuits. Work with a counselor / advisor to design a realistic schedule that dedicates adequate effort to be successful in college studies;
3. Know and follow the regulations and guidelines outlined in the Student Code of Conduct and Student Handbook;
4. Maintain respectful and appropriate behavior within and outside the classroom;
5. Ask for help when needed. Use all available resources and facilities provided by the College to enhance the learning experience;
6. Attend scheduled advising sessions, tutorials, and other appointments. Cancel or reschedule only with good reasons as early as possible;
7. Arrive prepared for tutorial sessions, bringing all needed materials (books, syllabi, rough drafts, calculators, assignment sheets, etc.).

C. Textbook Availability

A student of this institution is not under any obligation to purchase a textbook from a university-affiliated bookstore. The same textbook may also be available from an independent retailer, including an online retailer.

COLLEGE REQUIREMENTS:

While other exams are given at the discretion of the instructor, a final assessment is given at the end of each semester for each course. The Final Exam Schedule changes with each term and differs from normal class meeting dates and times. See the Final Exam Schedule in the Catalog/Schedule of Classes in the left hand navigation bar.

A student who must be absent from a final evaluation should petition that instructor for permission to postpone the evaluation. A student absent without permission from a final evaluation is graded "0" on the exam.

Incomplete Grades. The conditional grade of "I" may be issued to a student having a passing average on all completed coursework but for a justified reason, such as illness or death in the family or by providential hindrance, has been prevented from taking the final examination or completing other required coursework. The "I" becomes an "F" in one hundred twenty (120) calendar days from the end of the term unless the student completes the balance of the coursework with a performance grade of "D" or higher. Re-enrollment in the course will not resolve the "I." The student and faculty must fill out an Incomplete Contract, clearly defining the work remaining to be finished.

COLLEGE POLICIES

COLLEGE POLICIES:

A. All of the Alamo Colleges are tobacco free.

B. Alamo Colleges DPS Emergency Phone Numbers:

Emergency Phone (210) 485-0911

General Phone (210) 485-0099

Weather Phone (210) 485-0189 (For information on college closures)

D. Students are required to silence all electronic devices (e.g., pagers, cellular phones, etc.) when in classrooms, laboratories and the library.

Disability Access Statement – In accordance with the Americans with Disabilities Act (ADA) and Section 504 of the Rehabilitation Act, it is the responsibility of the student to self-identify with the campus Disability Services office. Only those students with appropriate documentation will receive a letter of accommodation from the Disability Services office. Instructors are required to follow only those accommodation and/or services outlined in the letter of accommodation. For further information, please contact the Disability Services office at (210) 486-3020 or visit the office located in the Palomino Center, Room 101. If you have specific needs, please discuss them privately with your instructor.

Module-by-Module Activities Schedule for C++ Students

Misc. Comments on assignments and other activities:

- Recommended (optional) end-of-chapter activities include answering multiple choice review questions, and "what will this code do" type of questions. This should help you with your exams.
- Check the exact assignment and exam due dates on the Canvas calendar
- Detailed assignment submission guidelines are given [here](#).
- [This document](#) briefly describes what kind of help you can expect with your assignments
- If you are using an older edition of the book, to avoid the pages discrepancies, use this [assignment document](#).
- Exams can not be made up. Please read the [make-up policy here](#).

Module Start date	Textbook Chapter	Assignments and Exams	Assignments - additional information
	1.6 - 1.7 [Intro - programming process, OOP, software engineering] 7 [arrays]	Download and install C++ compiler. Click for instructions Setting up your compiler (non credit assignment)	<p>You need to complete your first BioSig identification on the first day of class. This is very important. Students who miss this deadline, will be considered absent and dropped as per College policy.</p> <p>You also need to turn in your first assignment on time. Students who do not, will be dropped for non participation.</p>
	8 [sorting and searching arrays, algorithm efficiency]	A1 - Arrays "Sorting Benchmarks", pg. 488	<p>Use the following function prototypes: int bubbleSort(long [], int); int selectionSort(long [], int);</p> <p><i>Note: int in argument list is meant for array size. Function type is int because the</i></p>

			<p><i>function returns number of exchanges</i></p> <p>Turn in the following files: a1main.cpp program compile and run screenshots design document</p>
	9.1 - 9.6 [pointers, pointer arithmetic, initializing, comparing]		
	9.7 - 9.10 [pointers as function parameters, dynamic memory allocation, returning pointers]	<p>A2 - Pointers "Movie Statistics", pg. 539 (see previous problems instructions (8,9) in regards to median and mode functions)</p>	<p>Use the following function prototypes: double median(int *, int); int mode(int *, int); int *makeArray(int); void getMovieData(int *, int); void selectionSort(int [], int); double average(int *, int);</p> <p>Turn in the following files: a2main.cpp program compile and run screenshots design document</p>
	10 [character conversion, C-Strings, String lib]		
	11.1 -11.6 [ADT, structures, initializing, accessing, arrays of]	<p>A3 - Character and String Processing "Morse Code Converter", pg.591 (if the code seems to be getting really long, you are doing something wrong)</p>	<p>Use the following function prototypes: string toMorse(char);</p> <p>Turn in the following files: a3main.cpp program compile and run screenshots design document</p>
	11.7 - 11.12 [structure as args. ,returning a struct, pointers to structs, unions, enumeration]		
	12.1 - 12.5 [file operations, output formatting, passing file stream to functions, reading and writing, error testing]	<p>A4 - Structures "Drink Machine Simulator", pg.648</p>	<p>Use the following function prototypes: int getChoice(Drink []); void transaction(Drink [], int, double &);</p>

			<p>Turn in the following files: a4main.cpp program compile and run screenshots design document</p>
	<p>12.6 - 12.10 [multiple files, binary files, records from structures, random access files; opening file for I/O]</p>	<p>Midterm (to 11.12). Click here for review</p> <p>check Calendar for exact dates</p>	<p><i>Note: start working on A5 this Module! (you will see why, once you start)</i></p>
	<p>13.1 - 13.6 [OOP, classes, instances, private members, class specification, inline member functions]</p>	<p>A5 - Files Database management</p> <p>This assignment is not in your book. Access it from pdf assignment file.</p>	<p>"KEY" assignment</p> <p>Use the following function prototypes: void addRecord(fstream &); void viewRecord(fstream &); void changeRecord(fstream &);</p> <p>Turn in the following files: a5main.cpp program compile and run screenshots design document sample created file</p>
	<p>13.7 - 13.16 [constructors, passing arguments to constructors, destructors, overloading constructors, private member functions, arrays of objects]</p>	<p>A6 - OOP 1 "Circle Class", pg. 794</p>	<p>Turn in the following files: a6main.cpp program compile and run screenshots design document</p>
	<p>14.1 - 14.5 [instance and static members, member-wise assignment, copy constructors, operator overloading]</p>		
	<p>14.6 - 14.8 [object conversion, aggregation, class collaboration]</p>	<p>A7 - OOP 2 "Parking Ticket Simulator", pg.867</p>	<p>Turn in the following files: a7main.cpp program compile and run screenshots design document ParkedCar.h</p>

			ParkingMeter.h ParkingTicket.h PoliceOfficer.h
	15.1 - 15.4 [inheritance, protected members, base and derived class, redefining base class]		
	15.5 - 15.7 [class hierarchies, polymorphism, abstract base classes, pure virtual functions]	A8 - OOP 3 "Pure Abstract Base Class Project", pg. 944	Turn in the following files: a8main.cpp program compile and run screenshots design document Circle.h Rectangle.h BasicShape.h
		Final (12.1 - 15.7). Click here for review) check Calendar for exact dates	