DeVRY UNIVERSITY

Course Syllabus **Spring 2008**

Course Title:	Structured Programming with Lab (Compressed)
Course number:	COMP-122
Credit/Contact hour:	3-0-3
Course Dependency:	Prerequisite: ECET-100 **Required Co-requisite**: None
Class Schedule:	M-W-F 8:00 - 9:50am Room 238
Instructor:	Tom Wheeler E-mail: twheeler@devry.edu; alternate: tom.n0gsg@gmail.com Voice Mail: 816.941.0430 x5211
Office Hours:	M-W-F 11:00 - 11:50am Other times possible by appointment

Course Description

This course introduces structured design and programming techniques, as well as common tools to write, compile, run and debug programs written in a high-level programming language to solve a variety of engineering problems.

Textbooks and Materials

Textbook/s Title:	**Ed**	**Author**
C++ Programming from Problem Analysis to Program Design, (Thompson/Course Technology) (DeVry standard text)	**3rd**	**Malik, D.S.,**

Useful Resources

- *Topic notes* for each course section are posted on eCollege. These can help your understanding by complementing what is presented in the textbook.

- Practice examinations for the course can be accessed on the instructor's web site at:

  http://faculty.kc.devry.edu/twheeler/comp122

- It is recommended that you install Visual C .NET onto your own computer for home study, as this will facilitate completion of homeworks for this course. You may also use on-campus computers for homework completion if this option is not available to you.

Terminal Course Objectives (TCOs):

Following are the objectives for this course. Individual faculty, based upon their experience and expertise may add to these objectives to meet local campus needs. Any such additions will be communicated to the class. While the instruction remains focused in helping students, accomplishing these objectives is a shared responsibility of students and faculty. The outcomes of this course will depend upon the motivation and capabilities of the students, sufficient time allocation for studying, and the effectiveness of that effort.

DeVry University is committed to the continual improvement of its curriculum and instruction and to meet the needs of students and employers in a rapidly changing global economy. Students, faculty, and the university must all be actively involved to accomplish these objectives, as well as the objectives of this particular course.

Every class is to some extent a unique interactive experience, which may cause some variance within the stated objectives, in either content or level. Individual faculty, based on their experience and expertise, are encouraged to add objectives, as they deem appropriate, and to communicate these directly to the class. The outcomes of the course will depend on the design of the course, the quality of instruction, and the motivation and capabilities of the students, including time available for studying and the effectiveness of the effort.

1. Given a problem description, write a high-level language program that adheres to a set of published style rules for readability (indentation, placement of block delimiters, capitalization conventions for variable and function names, placement of useful comments, etc.).

2. Given a program specification that requires data to be entered, calculations to be performed and results to be displayed, develop a program that accomplishes the task. The design should be based on good practices of structured programming.

3. Given a program specification or algorithm that requires complex decision-making, construct different implementations of the design using different selection statements available in the high-level language, for example one with a sequence of ifs, another with nested ifs and a third with a switch.

4. Given a technical application such as a combinational logic circuit that includes AND, OR, XOR gates, construct a simulation of the circuit using a high-level language.

5. Given a program specification and pseudo-code design that involves multiple selection and repetition structures, write a high-level language program that implements the design.

6. Use the major components of an Integrated Development Environment (IDE) to enter, compile, run and debug high-level language source code.

7. Given a problem description and pseudo-code design of its solution, develop and document a test plan that adequately tests the implementation.

8. Demonstrate the use of arrays, nested loops, functions, call by value, call by reference or pointers.

9. Demonstrate the use of strings and formatted output, such as a table of input and calculated values, using tools available in the high-level language.

10. Given prescribed technical documentation guidelines, develop written reports and oral presentations with technical content.

EET/CET Program Objectives and Outcomes:

(Student competencies achieved at the time of graduation)

1. Conduct experiments involving electronic systems using modern test equipment, interpret test results and use them to improve products or methodologies.
   - 1.1. Performs Needs Analysis – define the problem
   - 1.2. States goals and objectives of the experiment
   - 1.3. Identifies resources to conduct experiment (parts, equipment, data sheets, etc.)
   - 1.4. Develops a procedure and collect data using modern test equipment
   - 1.5. Analyzes test results and draw conclusions.

2. Create, implement high-level and Assembly language programs in support of technical activities.
   - 2.1. Analyzes the problem logically
   - 2.2. Designs the solution
   - 2.3. Implements the solution
   - 2.4. Tests and debugs the software

3. For EET: Use the principles of science, mathematics, and engineering technology to design, implement, and evaluate hardware and software solutions to complex technical problems,
   - 3.1. Selects and defines a meaningful problem taking safety, ethical, social, economic, and technical constraints into consideration.
   - 3.2. Devises process to solve problem
   - 3.3. Applies appropriate knowledge of scientific, mathematical, and engineering design tools toward the design and analysis of problem solutions.
   - 3.4. Identifies key issues in designing and building a prototype
   - 3.5. Builds, tests and troubleshoots prototype
   - 3.6. Optimizes prototype with a commitment to quality, timeliness, and continuous improvement.

For CET: Use the principles of science, mathematics, software engineering, and engineering technology to design, implement, and evaluate software solutions to complex technical problems.
   - 3.1 Identifies a meaningful problem and defines preliminary solution specifications taking safety, ethical, social, economic, technical constraints, and user requirement into consideration
   - 3.2 Designs and implements appropriate data structures and algorithms
   - 3.3 Prepares a plan of action to implement the system
   - 3.4 Applies scientific, mathematical, software, and engineering design tools toward the design and analysis of problem solution
   - 3.5 Writes and tests readable and maintainable code
   - 3.6 Optimizes code with a commitment to quality, timeliness, and continuous improvement

4. Communicate effectively both orally and in writing.
   - 4.1 Communicates effectively in wiring
   - 4.2 Communicates effectively orally

5. Work effectively in a team environment.
   - 5.1 Exhibits good dialoguing skills
   - 5.2 As part of a small group project, when assigned roles, performs roles effectively

6. Apply applied research and problem-solving skills to support learning at DeVry as well as life-long personal and professional development.
   - 6.1 Recognizes the need to know information beyond one's own expertise and has the ability to gather and synthesize the necessary information into the solution of a problem
   - 6.2 Uses engineering problem-solving methodology in solving problems

7. Evaluate the broader effects of technology and to identify connections between technology and economics, politics, culture, ethical responsibility, social structure, the environment and other areas.
   - 7.1 Identifies linkages and causal relationships between technology and social, political, economic, cultural, and environmental conditions.
   - 7.2 Works effectively in diverse environments and adapts technical solution to solution a diverse audience
   - 7.3 Pursues technical work within guidelines for professional, ethical, and social responsibility

Class Policies and Procedures:

Attendance

Each student is required to attend every lecture and laboratory session in which he or she is enrolled. A swipe-card terminal (ATS) in each classroom is used to record attendance electronically. Students are responsible for arriving before class begins, sliding their identification card through the wall-mounted reader, and remaining for the duration of the course meeting. Students who are absent for two or more days should notify their Professor or assigned Academic Advisor in advance. Students who miss more than five (5) consecutive days of school are in violation of the DeVry attendance policy and will be dismissed. **Unexcused absence in excess of 5 hours will result in a reduction of the final course grade by one letter**.

Homework Expectations

Homework is due at the beginning of class (xx00 UTC). ***Late homework is not accepted unless mitigating circumstances are present (documentation will be required).*** Homework carries the weight of one major exam (100 points) in the course. Failure to do homework will do severe damage to your grade. (UTC=Universal Coordinated Time, or Standard World Time.)

Homework Performance Standards

- Unless specifically noted, all homework is to be done individually. If you need help solving a problem, it is acceptable to ask the instructor or a fellow student for assistance as long as you do your own work.

- For problems involving calculations, all work must be shown. If a numerical answer is obtained without doing a calculation, state clearly that this is the case. For example: "By inspection, the potential is 25 Volts."

- When showing work for numerical problems, all defining equations will be stated first. The last step in the problem will be substitution of values into the equations. For example:

  Given V = 20V and R = 5 Ohms, find the current I.

  $I = \frac{V}{R}$ (Comment: The defining equation, Ohm's law, is stated.)

  $I = \frac{20V}{5\Omega} = \underline{\underline{4A}}$ (Comment: Note that units are clearly displayed for the answer.)

- When a numerical answer is given, it must be boxed or underlined and have correct units attached.

- For program listings, your name must appear at the top of the listing. Each function must have a comment header stating the function name, purpose, arguments, and return conditions. Each major idea within the code must be properly commented.

- Programs that that lack appropriate division of functionality will receive a grade of zero (0). An example of such code would be a program performing all of its functionality within a single function such as `main()` when it would be more appropriate to divide the workload among several related functions.

(Homework Performance Standards are continued on next page)

Homework Performance Standards (Continued)

No credit will be given for any problems that have not been worked according to these instructions, or any additional instructions given by the instructor.

Make-Up Exams

No make up exams are given in COMP-122 for any reason.

Course Grading Standards

There are two major exams, an unspecified number of quizzes given at random intervals, various homework assignments, and a comprehensive final examination given in the last week of the course. Your grade will be determined as follows:

2 Major Exams	200 points (22% each, 44% total)
Quizzes/Homework	100 points (22%)
Final Exam	150 points (34%)
	450 points total for course (100%)

Note: No makeup exams are given. All examinations will be announced at least 1 week prior to administration. All students must take the final exam.

A final letter grade is to be awarded to each enrolled student in accordance with the 4.00 grading system shown below:

Letter Grade	***Percent of Total Points***	***Grade Points***
A	90 – 100%	4.00
B	80 – 89%	3.00
C	70 – 79%	2.00
D	60 – 69%	1.00
F	Below 60%	0.00

Academic Integrity Policy

Ideas and learning form the core of the academic community. In all centers of education, learning is valued and honored. No learning community can thrive if its members counterfeit their achievement and seek to establish an unfair advantage over their fellow students. The academic standards at DeVry are based on a pursuit of knowledge and assume a high level of integrity in every one of its members. When this trust is violated, the academic community suffers injury and must act to ensure that its standards remain meaningful. The vehicle for this action is the Academic Integrity Policy outlined in the *Student Handbook*.

The Academic Integrity Policy is designed to foster a fair and impartial set of standards upon which academic dishonesty will be judged. All students are required to read, understand, and adhere to these standards, which define and specify the following mandatory sanctions for such dishonest acts as copying, plagiarism, lying, unauthorized collaboration, alteration of records, bribery, and misrepresentation for the purpose of enhancing one's academic standing:

- The ***first recorded offense*** will result in the student receiving zero credit for the entire paper, exam, quiz, lab, homework assignment, or other graded activity in which the incident of academic dishonesty occurred. No partial credit may be given. Where the incident involved a graded assignment normally subject to a "drop" option, the student may not exercise that option.

- The ***second recorded offense*** will result in the student receiving a failing grade for the course in which the second offense occurs. The second offense need not be in the same course, program, or term as the first offense to invoke this sanction.

- The ***third recorded offense*** will result in the student being permanently expelled from the DeVry system. Again, the third offense need not be in the same course, program, or term as either the first or second offense to invoke the sanction.

Changes to Syllabus:

The contents of this syllabus are subject to change with appropriate notice to the students.

Weekly Course Schedule:

Week	Topics	TCO's	What's Due?	Reading Assignment
1	Introduction to C/C++ environment. Documentation of program code; shop standards. Development process.	1,6	Lab #1	Chapter 2 pages 29-113
2	C++ Data types and expressions; development of a simple application that utilizes user input; formatted output (printf); debugging introduction.	1,2	Lab #2, HW#1	Chapter 3 pages 115-166
3	Structured decision making (if-then, switch) and elementary loop structures; debugging. **Exam #1**	3,4	Lab #3	Chapter 4 pages 167-230; Chapter 5 pages 231-308
4	Functions and parameter passing without pointers; arrays.	5,8	HW#2	Chapter 6 - pages 309-344; Chapter 7 - pages 345-418
5	Pointers and references; pointers and arrays; passing parameters by pointer; significance of array names.	6,7	Lab #4	Text: pages 15-16,22, 196-198
6	Character arrays, strings, pointers (continued); formatting with sprintf and similar library functions. ; **Exam #2**	6.9	Lab #5	Handouts and Lecture Notes
7	Nested loops; fun with pointers; development of larger projects (multiple files and libraries)	8.9	HW#3	Chapter 9 - pages 473-550
8	**Wrap-Up & Final Exam**	ALL	Lab #6	NONE

* This schedule is tentative and subject to change to meet the needs of the class members.

Note 1: See "homeworks.doc" under eCollege "docSharing" for the homework problems.

Note 2: LAB due dates: Labs are due by the end of class on the Friday of each week shown on the schedule.

Note 3: HOMEWORK due dates: Homework is due at the beginning of the Friday class of each week.