

**Course** COSC 1550 - Computer Programming I

**Term** Fall Semester, 2008

**Instructor** Name: **Eng. Ioan Stelian VLAD, MS**  
Phone: **0650-2534025 or 01-260022748**  
Email: [vladi@webster.edu](mailto:vladi@webster.edu) or [isvlad2001@yahoo.com](mailto:isvlad2001@yahoo.com) or [i.s.vlad@direkt.at](mailto:i.s.vlad@direkt.at) or [i.s.vlad@iaea.org](mailto:i.s.vlad@iaea.org)

**Catalog Description** Teaches students programming as a systematic discipline and as a problem-solving tool and introduces students to the C++ programming language. Acquaints students with fundamental concepts of computers, information processing, algorithms, and programs.

**Prerequisites** None.

**Learning Outcomes**

- a) The student will learn programming concepts and will acquire skills needed to program in the C++ computer language.
- b) The student will be able to write short programs in pseudo code and in the C++ programming language.
- c) The student will be able to read a simple C++ program and predict the outcome and output of the program.

**Textbook and other Materials**

*Mandatory*  
Tony Gaddis: Starting Out with C++ (Early objects) 5<sup>th</sup> Edition;  
Scott/Jones Inc., ISBN: 0-321-38348-6

*Supplemental*  
Bjarne Stroustrup. AT&T: The C++ Programming Language; Third Edition;  
Addison-Wesley, ISBN 0-201-88954-4  
D. S. Malik: C++ Programming-From Problem Analysis to Program Design; Thomson Course Technology, ISBN 0-619-16042-X

**Grading** Grading is the standard Webster grading scheme as follows:  
**A:** 93-100, **A-:** 90-92, **B+:** 87-89; **B:** 83-86; **B-:** 80-82; **C+:** 77-79; **C:** 73-76; **C-:** 70-72; **D+:** 67-69; **D:** 60-67; **F:** <60

	Activity category	% of grade
a)	Class Participation and reading of assigned text	10%
b)	Mid-Term Test	20%
c)	Final Exam	30%
d)	Preparation & Class Discussion of Case Studies	20%
e)	Homework Assignments	20%

While there are no prerequisites for this course, students who would like to obtain a major in one of the Information Technology programs are advised that the following course (COSC 1560 - Computer Programming II) has a prerequisite of a grade B or better on this course.

## **Activities**

Each theoretical explanation will be followed by a practical session, where the concepts are discussed and programmed; group discussions and experience sharing are encouraged; new ideas regarding solutions to problems/assignments/demos are most welcome.

Each class will start with a five minutes review of the topics covered in the previous class. During this time questions may be asked as well as short quizzes may be given to test that reading assignment were completed as required.

Note: All appendices referred below can be found on the book's accompanying CD in the folder Book\_Resources\Appendices as PDF documents.

## **University Policies**

### **Cheating**

Any student found cheating on any part of the course work (homework assignment, term paper, quiz, exam, etc.) will automatically be given an "F" for the course. Any student found to be helping another student to cheat will likewise be given an "F" for the course. In each case, the student will also be put on disciplinary probation for the remainder of her or his stay at Webster. If the student is found to have engaged in cheating a second time, s/he will be automatically expelled from the University.

### **Attendance**

Class attendance is obligatory. Instructors will take attendance at each class session and reduce a student's course grade by 1/3 of a letter grade for each full week of class missed. Any student who misses four or more full weeks of class will automatically be given an "F" for the course.

## **Course Policies**

During the course there shall be 7 homework assignments. Each contributes equally to the weight of point e) above. As a general rule, homework results should be presented in the time requested (generally one week); delays up to one week are acceptable with a mark penalty of up to 25%. Most homework assignments require that their resulting C++ code files be found in the student's folder resident on the Webster server in order to consider them fully closed. Penalties for missing files (entirely or in part) and/or files named other than required (making follow-up difficult) might range from 1 to 5 %. The instructor may choose at random one or more results from current homework for extensive verification. In this process, student source code will be compiled, an executable version will be created and run. Results should match those presented by the student, otherwise homework will not be accepted. For information and to enable student workload planning, the approximate moment when an assignment shall be distributed and shall be due is provided together with the reading assignments.

The course shall be a Web-enhanced course allowing students access to class material and feedback of current results. To reduce the paper work, be environmentally friendly and allow for best feedback to students, most of the results need to be printed in PDF and shall be submitted in Web CT. Detailed instructions shall be provided in due course.

Mid-term as well as final exam will contain a number of questions and small problems.

Each of these test items is valued 0 to 100%. The item contributes to the

calculation of the test's final score with the weight given by the section (A to D) in which it is listed. This weight may vary from two (for very simple questions) to five or six (for the programming exercises).

Some questions have multiple answers and at least one of them is correct. If more than one answer is valid, the 100% allocated to the question is equally divided between the number of correct answers. The number and difficulty of the exam items are chosen to enable a well-prepared student to finish in about 75% of the allocated examination time, allowing for review and style changes.

Successful completion of the course requires that the assigned text be read in advance of the session in which these are listed as Assignment.

## Weekly Schedule

### Week 1 *Class 1*

Introduction to Computers and Programming

Reading assignments: Textbook, Chapter 1, Pages 1 - 17.

### *Class 2*

Introduction to Computers and Programming: Numbers and numeration bases. Computer representation of numeric values. Number arithmetic and logic.

Reading assignments: None.

Assignment #1 distributed.

### Week 2 *Class 1*

Introduction to Computers and Programming: The Programming process.

Pseudo-code and flowcharting.

Reading assignments: Textbook, Chapter 1, Pages 18 - 28 and Appendix O (see CD).

### *Class 2*

Introduction to C++: C++ program parts and data types.

Reading assignments: Textbook, Chapter 2, Pages 29 - 61 and Chapter 1, Pages 13 - 15.

Assignment #1 due.

### Week 3 *Class 1*

Introduction to C++: Size of Data types. Scope of variables. Programming style.

Introduction to the MS Visual C++ and Borland Builder 6.0 Interactive Development Environments. Reading assignments: Textbook, Chapter 2, Pages 61 - 77, Appendices J, K and L (see CD) and instructor provided document.

### *Class 2*

Expressions and Interactivity: Expressions. Conversion. Assignments.

Reading assignments: Textbook, Chapter 3, Pages 79 - 101 and Appendix B, Page 1125.

Assignment #2 distributed.

Week 4 *Class 1*

Expressions and Interactivity: Constants. Input. Output. Mathematical library functions.  
Reading assignments: Textbook, Chapter 3, Pages 101 - 133.

*Class 2*

Expressions and Interactivity: More on Math.Library Functions. Practical examples (e.g. Cartesian to Polar coordinates interchange).

Reading assignments: Textbook, Chapter 3, Pages 135 - 165 and Appendix G (`math.h` part). *Note*: 3.12-Introduction to Files (pages 138 - 145) subject of COSC 1560 course.

Assignment #2 due.

Week 5 *Class 1*

Practical class: Developing our first C++ program.

Reading assignments: Textbook, Chapter 1 to 3, Pages 1 - 165 and Appendices J, K and L.

*Class 2*

Making Decisions: Operators, Conditions, If statement. Complex logical conditions.

Reading assignments: Textbook, Chapter 4, Pages 167 - 192.

Assignment #3 distributed.

Week 6 *Class 1*

Making Decisions: Menus and nested menus. Logical operators. User input validation.

Reading assignments: Textbook, Chapter 4, Pages 192 - 215.

*Class 2*

Making Decisions: Comparing characters and strings. Conditional operator. Switch Statement.

Reading assignments: Textbook, Chapter 4, Pages 215 - 254.

*Note*: 4.15-Testing for File Open Errors (pages 235 - 236) subject of COSC 1560 course.

Week 7 *Class 1*

Looping: Introduction to loops. Counters. Running Totals. Sentinels.

Reading assignments: Textbook, Chapter 5, Pages 255 - 284.

*Note*: 5.7-Using a Loop to Read Data from a File (pages 271 - 274) subject of COSC 1560 course.

Assignment #4 distributed. Assignment #3 due.

*Class 2*

Looping: Nested loops. Terminating and continuing loops. Loops and user input validation.

Reading assignments: Textbook, Chapter 5, Pages 284 - 312.

Week 8 *Class 1*

Practical class: Developing together a second C++ program and review of the topics covered in weeks 1 to 7 as preparation of mid-term test.

*Class 2*

Mid-Term Test. Test will include questions from all topics covered until week 8.

Week 9 *Class 1*

Mid-Term Exam evaluation: Review of results and discussion of the correct solutions.

Reading assignments: None.

Assignment #5 distributed. Assignment #4 due.

*Class 2*

Functions: Defining a function. Function prototype. Function parameters and arguments.

Reading assignments: Textbook, Chapter 6, Pages 313 - 332.

Week 10 *Class 1*

Functions: Return statement. Local and global variables. Default function arguments.

Reading assignments: Textbook, Chapter 6, Pages 332 - 356.

*Class 2*

Functions: Reference variables. Overloaded functions. The EXIT function. Stubs and drivers.

Reading assignments: Textbook, Chapter 6, Pages 357 - 390 and Appendix J (see CD).

Assignment #5 due.

Week 11 *Class 1*

Arrays: Definition. Accessing. Initialization. Processing array contents.

Reading assignments: Textbook, Chapter 8, Pages 477 - 502.

*Note*: 7-Introduction to Classes and Objects (pages 391 - 476) subject of COSC 1560 course.

*Class 2*

Arrays: Parallel arrays. Arrays as function arguments. Multi-Dimensional arrays. Arrays of strings.

Reading assignments: Textbook, Chapter 8, Pages 502 - 528.

Assignment #6 distributed.

Week 12 *Class 1*

Arrays: Matrix multiplication. Array case studies.

Reading assignments: Textbook, Chapter 8, Pages 540 - 542 and 555 - 566.

*Note*: 8.11-Vectors (pages 528 - 540), 8.13-Arrays of Structures (pages 543 - 546) and 8.14-Arrays of Class Objects (pages 547 - 555) subject of COSC 1560 course.

*Class 2*

Searching Arrays: Linear Search and Binary Search.

Reading assignments: Textbook, Chapter 9, Pages 567 - 581.

Assignment #6 due.

Week 13 *Class 1*

Sorting Arrays: Bubble and Selection sort.

Reading assignments: Textbook, Chapter 9, Pages 581 - 596 and 601 - 606.

*Note:* 9.5-Sorting and Searching Vectors (pages 596 - 601) subject of COSC 1560 course.

*Class 2*

Further Sorting Array algorithms: Shaker sort. Insertion sort. Binary straight Insertion sort, Insertion sort with diminishing increment + third hands-on content to be determined week 12.

Reading assignments: Documents will be distributed to students in week 12.

Assignment #7 distributed.

Week 14 *Class 1*

Pointers: Address of a variable. Arrays and Pointers. Pointer arithmetic.

Reading assignments: Textbook, Chapter 10, Pages 607 - 621.

*Class 2*

Pointers: Comparing pointers. Dynamic memory allocation. Pointers as function parameters and arguments. Sorting using pointers.

Reading assignments: Textbook, Chapter 10, Pages 621 - 656.

*Note:* 10.10-Pointers to Structures and Objects (pages 635 - 640) and 10.11-Selecting Members of Objects (pages 641-642) subject of COSC 1560 course.

Assignment #7 due.

Week 15 *Class 1*

Review and final exam preparation

*Class 2*

Final Exam. Test for final exam will include 25% questions targeting topics covered in weeks 1 to 8 and 75% questions targeting topics taught during weeks 9 to 15.

**Additional  
Information**

Syllabus Approved: 2008-6-05

Brian Lewis, Computer Science Department Head