

SYLLABUS CMPSC 201 Spring 2014
Programming for Engineers with C++
(Fulfills requirement for many Engineering and
Science majors)

Instructor: PRIYA ANAND . B.E., M.S.

Office: 267 Hawthorn

Phone: 940-3105

Electronic-Mail ID: AXG36@PSU.EDU

Office Hours:

267 Hawthorn: Tues & Thurs -1:15pm – 2/30pm.

Class Hours: 259 Hawthorn: Tues & Thurs 2:40am-3:55pm.

Faculty Staff Assistant:

Stephanie Tanner Room: 203 Hawthorn. Phone: 949-5258

Email Address: SAW187@PSU.EDU

Course Overview: Programming for Engineers with C++ Development & implementation of algorithms in a procedure-oriented language, with emphasis on numerical methods for engineering problems. A student may receive credit for only one of the following courses: CMPSC 101, 102, 200, 201, or 202. Prerequisite: MATH 140. Concurrent: MATH 141.

Textbook: *Brief Version of Introduction to Programming with C++* by Y. Daniel Liang, published by Prentice Hall, 2007. Plus *Introduction to MATLAB 7* by Delores M. Etter, David C. Kuncicky with Holly Moore, published by E Source, Prentice Hall.

Course Objectives: CMPSC 201 introduces the fundamental concepts of procedural programming. Topics include data types, control structures, procedures, arrays, files, and the mechanics of running, testing and debugging a program with a language sensitive editor and an interactive debugging environment. Numerical analysis techniques may be used to illustrate these topics.

This course uses C++ as the procedural computer language to prepare engineering and science majors to meet immediate demands in solving computational problems. This course will also introduce students to the MATLAB interface and problem solving with this common program.

Primary Course Outcomes:

Upon completion of the course, students should possess the following skills:

- **Problem Definitions:** Given a problem, students can create a functional description of the problem including input and output specifications, and processing steps to obtain the output from the input. This description should be added to code as comments.
- **Problem Synthesis:** Given a problem, students should be able to decompose the problem into a sequence of single-purpose functions that are highly cohesive and loosely coupled.
- **Data Types:** Given a functional description of a problem, students can select appropriate data types (including primitive data types, structures, and arrays) for storing the input and output values. Declare, define, and initialize one and multi-dimensional, simple data type arrays of a fixed size. Demonstrate the ability to read from and write to an arbitrary array element using array indices. Demonstrate an ability to process the entire array, one element at a time, performing both read and write operations.
- **Assignment Statements:** Given a mathematical expression, students can translate this expression into equivalent syntactically correct programming statements.
- **Style/Documentation:** Given a problem, students can write code that conforms to a programming style specified by the instructor.
- **Control Structures:** Given a problem, students can select and implement the appropriate control structure (if/else, switch, for-loop, while-loop, and/or do while-loop). Given a problem and a specific control structure, students can implement that control structure to solve the problem. Given the pre/post conditions, students can write functions utilizing various return types, pass by reference parameters, and/or pass by value parameters as appropriate.
- **Code Analysis:** Given an unfamiliar code segment and specific inputs, students can analyze the code and predict the result.
- **Testing and Debugging:** During program development students can utilize basic testing methodologies and debugging tools such as stubs, drivers, and integrated debuggers to identify fault points and possible error conditions.
- **File Input and Output:** Given a problem, a student can develop input/output modules to read data from the standard input or a text file and write data in a prescribed format either to the standard output or to a text file.

Required Topics:

- Introduction, overview of computers (including operating system, code generation). (~1.5 hrs)
- Algorithms, designs, programs, preprocessor directives, cout object. (~1.5 hrs)
- Variables, declarations, data types (including data representation), assignment statements, cin object, arithmetic operations. Precedence of arithmetic operations, exponents, type coercion, type casting, define, compound operators, incrementing and decrementing, string class. (~1.5 hrs)
- Formatting input and output. Simple file Input/Output. (~3 hrs)
- Relational operators, single branched if, two-branched if else. Multi-branched if (if/elseif), nested ifs, boolean flags. Logical operators, precedence of relational and logical operators. (~3 hrs)
- Switches, comparing strings, conditionals. (~1.5 hrs)
- Concept of loops/repetition, while loop, do-while. For loops, common problems, rewriting as other loop types. Break, continue, nested loops. (~3 hrs)
- Concept of functions, passing parameters, returning values, Implementation. (~3 hrs)
- Reference vs. value parameters. More function examples, stubs, drivers, exit function. (~3 hrs)

- Arrays, what they are, storage of arrays, how are they declared, using 1-D arrays. (~3 hrs)
- Searching 1-D arrays, sorting 1-D arrays, and passing arrays to functions. (~3 hrs)
- Multidimensional arrays. Character testing, character conversion, storage of c-strings. (~3 hrs)
- Structures and records with fields of different data types. (~3 hrs)
- Introduction to MATLAB interface and using MATLAB to solve simple engineering problems. (~6 hrs)

Professional Component:

CMPSC 201 is designed to aid in the professional development of engineers by developing skills in problem solving, critical thinking, algorithm design, and program implementation. Although C++ will be used to demonstrate these skills, many of the concepts can be applied to general engineering problems and writing programs in other languages as well.

Course Points will total 1000 and be reported on ANGEL. The grading scale follows:

Grade of	Point range	
A	= 900-1000;	
A-	= 880-899;	
B+	= 850-879;	
B	= 800-849;	
B-	= 780-799;	{ below this grade, the attrition rate is ~ 99% }
C+	= 750-779;	
C	= 700-749;	{ minimal requirement for CMPSC and CMPEN majors }
D	= 600-699;	
F	< 600.	

☺ CLASS ATTENDANCE: Required. Computer demonstrations of problem solving occur in lab via large screen display. Programming assignments are introduced by the instructor with student engaged discussion about detailed specifications that culminate in partial solutions which may be used by students for lab projects. Students not in lab for discussion of programming specifications are individually responsible for the material missed

☺ COURSE ETHICS: Students are responsible for completing individual assignments on your computing devices. Students using work or claiming work of others is unethical and a form of plagiarism. **Working with others or receipt of assistance must be cited in the documentation of programs, specifically at the beginning of the file comment lines of your project module.** Any use of design or code from classmates, tutors, friends, textbook references, web references, etc., must be documented. Unethical practices are addressed in Altoona College's and Computer Science Department's Policies.

<http://www.aa.psu.edu/academic/integrity.htm> See also

PSU Course Policies provided later.

Sample **documentation of programs with receipt of assistance** follows:

```
//Programmer: Instructor
//Date Due: 1/17/12
//Page 77 Program Challenge 6. Annual Pay--as Lab3.zip assignment.
//Assistance with cout statements provided by Rob the tutor on 1/15/12.
```

PENN STATE COURSE POLICIES

Academic Integrity Policy

According to the Penn State Principles and University Code of Conduct: *Academic integrity is a basic guiding principle for all academic activity at Penn State University, allowing the pursuit of scholarly activity in an open, honest, and responsible manner. In accordance with the University's Code of Conduct, you must not engage in or tolerate academic dishonesty. This includes, but is not limited to cheating, plagiarism, fabrication of information or citations, facilitating acts of academic dishonesty by others, unauthorized possession of examinations, submitting work of another person, or work previously used without informing the instructor, or tampering with the academic work of other students.*

Any violation of academic integrity will be investigated, and where warranted, punitive action will be taken. For every incident when a penalty of any kind is assessed, a report must be filed. This form is used for both undergraduate and graduate courses. This report must be signed by both the instructor and the student, and then submitted to the College representative.

Affirmative Action & Sexual Harassment Policy

The Pennsylvania State University is committed to a policy that all persons shall have equal access to programs, facilities, admission, and employment without regard to personal characteristics not related to ability, performance, or qualifications as determined by University policy or by Commonwealth or Federal authorities. Penn State does not discriminate against any person because of age, ancestry, color, disability or handicap, national origin, race, religious creed, gender, sexual orientation, or veteran status. Direct all inquiries to the Affirmative Action Office.

Americans with Disabilities Act

CSE welcomes persons with disabilities to all of its classes, programs, and events. If you need accommodations, or have questions about access to buildings where CSE activities are held, please contact us in advance of your participation or visit. If you need assistance during a class, program, or event, please contact the staff or faculty member in charge of your CSE department.

An Invitation to Students with Learning Disabilities

It is Penn State's policy to not discriminate against qualified students with documented disabilities in its educational programs. If you have a disability-related need for modifications in your testing or learning situation, your instructor should be notified during the first week of classes so that your needs can be accommodated. You will be asked to present documentation from the Office of Disability Services that describes the nature of your disability and the recommended remedy. You may refer to the Nondiscrimination Policy in the Student Guide to University Policies and Rules.