Angelina College Science and Mathematics Division
COSC 1336: Programming Fundamentals
Instructional Syllabus - Spring 2017

I. BASIC COURSE INFORMATION

A. Course Description *(as stated in the bulletin, including necessary pre-requisite courses, credit hours)*

Computer Science - COSC 1336 - Programming Fundamentals I. Introduces the fundamental concepts of structured programming and provides a comprehensive introduction to programming for computer science and technology majors. Topics include software development methodology, data types, control structures, functions, arrays, and the mechanics of running, testing, and debugging. This course assumes computer literacy. This course is included in the Field of Study Cirriculum for Computer Science. Recommended Prerequisite: COSC 1315

B. Intended Audience

The intended audience is any student needing the fundamentals of structured programming, including but not limited to students preparing for the study of computer science.

C. Instructor -

Name: **Bill Fisk** Office: S112

Phone: (936) 633-5461 email: bfisk@angelina.edu

Office Hours:

Day	Class Hours	Office Hours
Monday	11:25-12:45pm	9:00-11:15; 1:30-4:00pm
Tuesday	8:00-9:20; 9:30-10:50, 11:25-12:45pm, 1-2:00p	2:00-4:00pm
Wednesday	11:25-12:45pm, 1-2:00pm (lab)	9:00-11:15; 2:00-4:00pm
Thursday	8:00-9:20; 9:30-10:50, 11:25-12:45pm	1:00-4:00pm
Friday	No classes	9:30-11:30am by appointment only

II. INTENDED STUDENT OUTCOMES:

A. Core Objectives Required for this Course

1. **Critical Thinking:** to include creative thinking, innovation, inquiry, and analysis, evaluation and synthesis of information..
2. **Communication:** to include effective development, interpretation and expression of ideas through written, oral, and visual communication
3. **Empirical and Quanitative Skills:** to include the manipulation and analysis of numerical data or observable facts resulting in informed conclusions. .

B. Learning Outcomes for all Sections - (Found in the Lower-Division Academic Course Guide Manual of the Texas higher Education Coordinating Board)

1. Analyze and explain the behavior of simple programs involving the fundamental programming constructs.
2. Modify and expand short programs that use standard conditional and iterative control structures and functions; choose appropriate conditional and iteration constructs for a given programming task.
3. Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions.
4. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
5. Describe the mechanics of parameter passing and demonstrate the difference between call-by-value and call-by-reference parameter passing.
6. Discuss the importance of algorithms in the problem-solving process, identify the necessary properties of good algorithms, and create algorithms for solving simple problems.
7. Use pseudocode or a programming language to implement, test, and debug algorithms for solving simple problems.

8. Discuss the representation and use of primitive data types and built-in data structures.
9. Explain the reasons for using different formats to represent numerical data.
10. Explain basic concepts of secure programming functions.
11. Discuss the properties of good software design.
12. Describe the phases of program translation from source code to executable code and the files produced by these phases; explain the software life cycle and its phases, including the deliverables that are produced.
13. Identify and describe the properties of a variable such as its associated address, value, scope, persistence, and size.
14. Explain how abstraction mechanisms support the creation of reusable software components.

III. ASSESSMENT MEASURES:

A. Assessments for the Core Objectives:

1. **Critical Thinking - Students** will use logic skills to develop an algorithm to solve a problem on the computer using a computer programming language. These programs will be written to accommodate different sets of data for different scenarios. Should logic errors occur in the programs, the students will dissect and debug their programs to determine those errors and resolve them.
2. **Communication:** Students will employ interactive prompts in their programs to solicit responses from the user (of that particular program). Offer short oral presentaitions on setting up various alogorithms.
3. **Empirical and Quanitative Skills** - Students will demonstrate logical processes for adding numbers in the binary numbering system, for converting a decimal number to a binary number, a binary number to a decimal number, a hex number to a binary number, a binary number to a hex number, and a decimal number to a Base-8 number (Octal).

B. Assessments for Course Learning Outcomes:

1. The student's ability to develop an understanding of the use and function of microcomputers will be assessed through monitoring classroom discussions and through quiz and test questions that pertain to these topics.
2. The student's ability to describe the development and creation of programming constructs using a programming language will be assessed through monitoring classroom discussions and through quiz and test questions pertaining to these topics.
3. The student's exploration and use of a high level programming language will be assessed through monitoring student's laboratory and out-of-class programming assignments and through test questions pertaining to this topic.
4. The student's ability to interpret error messages and debug a program will be assessed through monitoring students's laboratory and out-of-class programming assignments.
5. The student's ability to convert a descriptive problem into a complete and executable Object code using logical methods will be assessed through monitoring student's laboratory and out-of-class programming assignments and through test questions pertaining to this topic.

IV. INSTRUCTIONAL PROCEDURES:

A. Methodologies common to all sections

This course will be taught using a combination of lectures, discussions and programming examples. Lab time will be used for clarification of programming statements and procedures. The amount of time spent using any one technique will vary as determined to be most appropriate by the instructor.

B. Methodologies determined by the instructor – N/A

V. COURSE REQUIREMENTS AND POLICIES:

A. Required Textbooks, Materials, and Equipment –

1. Starting Out With C++, Brief Version by Tony Gaddis (7th edition)
2. **Specific equipment required of all students.**

Two flash drives (jump drives) and a flowchart template. You must be able to dedicate the flash drives to this course.

3. **Additional text(s) and supplementary materials for individual instructor** – N/A
4. **Specific equipment required by the individual instructor** – N/A

B. Assignments *(Appropriate due dates, schedules, deadlines)*

1. **Programming Assignments**: Up to five major programs with varying point value may be assigned during the semester. All programming assignments are due by 4:00 on the day the next program is assigned and will not be accepted late. All programming assignments should be stored on a jump drive to be turned in, and backed up to the network Drive K. If your jump drive fails and your assignment is not on Drive K, you will lose at least of 50% of the assignment's value
2. **Laboratory Assignments**: Up to ten lab programs may be assigned, each counting a maximum of 10 points. These programs will be assigned during lab hours and will be due by 1:10 the next day (following lab) and will not be accepted late. You will turn in a jump drive with lab programs, but ***do backup*** all your data file, source code, and executable code to Drive K. Organize your Drive K, so that labs and programs are easily identified.
3. **Worksheets** will be assigned as needed and will be worth a maximum of 10 points. These assignments are due by 4:00 on the class meeting after they are assigned and will not be accepted late.
4. **Quizzes** are not announced in advance and cannot be made up. Each quiz is worth 10 points.
5. No additional specific assignments, other than those previously discussed, are required by this instructor.

C. Course Policies – *(This course conforms to the policies of Angelina College as stated in the Angelina College Handbook.)*

Academic Assistance – If you have a disability (as cited in Section 504 of the Rehabilitation Act of 1973 or Title II of the Americans with Disabilities Act of 1990) that may affect your participation in this class, you should see Sellestine Hunt Associate Dean of Student Services, Student Center, Room 200. At a post-secondary institution, you must self-identify as a person with a disability; Ms. Hunt will assist you with the necessary information to do so. To report any complaints of discrimination related to disability, you should contact Mr. Steve Hudman, Dean of Student Affairs, in Student Center, Room 101, and (936) 633-5292 or by email **shudman@angelina.edu**.

Attendance – A student **may** be dropped after three consecutive or four cumulative absences. It is the student's responsibility to attend classes, participate in class discussions, and complete required work on time. Should the student decide to withdraw from the course, it is the student's responsibility to initiate and complete the drop process. The last day to drop or withdraw is April 4th.

Computer Lab Hours in S110:

Monday:	8:00 - 4:00
Tuesday:	8:00 - 4:00
Wednesday:	8:00 - 6:00
Thursday:	8:00 - 4:00
Friday:	8:30 - 3:30

Computer Laboratory Rules:

- Students working on Science & Mathematics Division assignments have priority on use of computers.
- Children and other guests are not allowed.

- NO food, drink, or tobacco products are allowed.
- Use of cell phones for conversation or texting is prohibited in the LAB.
- If you get a call on your cell phone, TAKE the conversation into the Hallway
- Use of the computers in any of the laboratories in the Science & Mathematics Division implies acceptance of the Computer Use Policy as posted.

Additional Classroom Policies Established by the Individual Instructor –

- Conferences outside of class are available during office hours or by appointment.
- **No** eating, drinking, or use of tobacco products is allowed.
- Children and other guests are not allowed.
- **NO makeup tests** are authorized except in case of emergency. A student is required to contact the instructor about a possible makeup exam as soon as possible, preferably on or before the date of the original exam. The final exam will replace any one missed test or the lowest of the scheduled tests during the semester.
- Beepers, cellular phones, and similar electronic devices are not appropriate in the classroom. When you come to class, please **turn them off and put them away**.
- Students are expected to participate in the instruction through courteous, relevant comments and questions during class and behavior that interferes with the learning environment is not tolerated.
- **Any student or students caught cheating (plagiarism, collusion, copying, etc.) on an exam or an assignment will receive a zero for that exam or assignment.**

VI. COURSE CONTENT:

A. Required Content/ Topics - *(as required by the individual Instructor)*

Main topics covered include:

Introduction to Computers and Programming (Chapter 1)
Introduction to C++ (Chapter 2)
Expressions and Interactivity (Chapter 3)
Making Decisions (Chapter 4)
Loops and Files (Chapter 5)
Functions (Chapter 6)
Arrays (Chapter 7)
Searching and Sorting Arrays (Chapter 8)
Characters, C-Strings, and More About the string Class (Chapter 10)

A Student Course Outline is attached which includes assignments and test dates. *There may be variations depending on circumstances, but it serves as a general guide for preparing for class.*

B. Additional Content - Other topics and material from additional chapters may be introduced.

VII. EVALUATION AND GRADING:

A. Grading Criteria – Grades are determined by numeric scores on the following written components:

Programs (Total of 200 points or 28.6%) The format of each program will vary, but each will contain specific instructions for the problem to be solved. The ability of the student to use the C language and follow the syntax rules along with solving the original problem will be reflected in the grade for each assignment. Programs will be due by 4:00 on the due date. Flash drives will be turned in with program assignments.

Laboratory Programs, Quizzes, and Worksheets (Total of 100 points or 14.3%) The format for each laboratory program will vary, but each will contain specific instructions for the problem to be solved. The ability of the student to use the language and complete the problem assigned will be reflected in the grade for each program. The requirements for each worksheet will be given as part of the assignment and the ability of the student to complete those requirements will be reflected in the grade for the worksheet. Lab programs will be due by 1:10 pm on the Wednesday following the assignment of the laboratory program. Flash drives will not be turned in with lab programs.

Exams (Total of 300 points [100 points each] or 42.9% [14.3% each])) The material covered is given on the class schedule attached to the end of this syllabus. There are no make-up exams.

Final Exam (100 points or 14.3%) This is a comprehensive exam.

A copy of your grades and current average will be distributed near mid-semester (after Exam 2) and near the end of the semester (after Exam 3) to inform you of your progress and status, and to allow verification that your grades have been accurately recorded.

B. Determination of Grade (*assignment of letter grades*) –

A	90 - 100
B	80 - 89
C	70 - 79
D	60 - 69
F	Below 60

VIII. SYLLABUS MODIFICATION:

The instructor may modify the provisions of the syllabus to meet individual class needs by informing the class in advance as to the changes being made.

FYI: Angelina College's campus security is available 24 hours a day by contacting 936-676-2563. Please use this number only as necessary for security issues.

Spring 2017 COSC 1336 Course Outline and Schedule			
Date		**Topics**	**Assignment**
Jan 18	1	Introduction to course; History of C; 1.1 Why Program?; 1.2; Computer Systems	
Lab #1		Conversion of Bases Review; Practical Logic Steps	Logic #1 [2.5]
Jan 23	2	1.3 Programs & Programming Languages; 1.4 What is a Program Made Of?; 1.5 Input, Programming, Output; Class Logic Development	
Jan 25	3	1.6 The Programming Process; Procedural and Object-Oriented Programming; Logic Development #2	Logic #2 [2.5]
Lab #2		Logic #3: "Animate in Alice"	Logic #3 [2.5]
Jan 30	4	Logic Development #4	Logic #4 [2.5]
Feb 1	5	2.1 The Parts of a C++ Program; 2.2 The cout Object; 2.3 The #include Directive; 2.4 Variables and Literals; 2.5 Identifiers	
Lab #3		Conversion of Negative Integers	Base Wks [10]
Feb 6	6	2.6 Integer Data Types; 2.7 The char Data Type; 2.8 The C++ string Class; 2.9 Floating-Point Data Types; 2.10 The bool Data Type	
Feb 8	7	3.7 Formatting Output	Program #1 [20]
Lab #4		Introduction to the Network and C++ Compiler	Lab #4 [10]
8 Feb 13	8	2.11 Determining the Size of a Data Type; 2.12 Variable Assignments and Initialization; 2.13 Scope; 2.14 Arithmetic Operators; 2.15 Comments; 2.16 Named Constants	
Feb 15	9	2.17 Programming Style; 2.18 Standard and Pre-Standard C++; 3.1 The cin Object; 3.2 Mathematical Expressions	Program #2 [40]
Lab #5		Input and Output	Lab #5 [10]
Feb 20	10	**Exam #1: Chapter 1 & 2**	
Feb 22	11	3.4 Overflow and Underflow; 3.5 Type Casting; 3.6 Multiple & Combined Assignment; 3.8 Working with Characters and string Objects; 3.9 More Mathematical Library Functions; 3.10 Focus on Debugging; 3.11 Focus on Problem Solving	
Lab #6		4.1 Relational Operators; 4.2 The if Statement; 4.3 Expanding the if Statement; 4.5 Nested if Statements; Work on Program #2	
Feb 27	12	4.6 The if/else Statement; 4.7 Flags; 4.8 Logical Operators; 4.9 Checking Numeric Ranges with Logical Operators	
Mar 1	13	4.10 Menus; 4.11 Validating User Input; 4.12 Comparing Characters and Strings; 4.13 The Conditional Operator; 4.14 The switch Statement	Program #3 [40]
Lab #7		Decisions	Lab #7 [10]
Mar 6	14	5.1 The Increment and Decrement Operators; 5.2 Introduction to Loops: The while loop; 5.3 Using the while Loop for Input Validation; 5.4 Counters	
Mar 8	15	5.5 The do-while Loop; 5.6 The for Loop; 5.7 Keeping a Running Total; 5.8 Sentinels; 5.9 Deciding Which Loop to Use	
Lab #8		Work on Program #3	
		SPRING BREAK	
Mar 20	16	5.10 Nested Loops; 5.11 Using Files for Data Storage; 5.12 Breaking and Continuing a Loop; Review for Exam #2	
Mar 22	17	**Exam #2: Chapter 3, 4, and 5**	
Lab #9		Loops	Lab #9 [10]

Spring 2017 COSC 1336 Course Outline and Schedule			
Date		**Topics**	**Assignment**
Mar 27	18	6.1 Modular Programming; 6.2 Defining and Calling Functions; 6.3 Function Prototypes	Program #4 [50]
Mar 29	19	6.4 Sending Data into a Function; 6.5 Passing Data by Value; 6.6 Using Functions in a Menu-Driven Program; 6.7 The return Statement; 6.8 Returning a Value from a Function; 6.9 Returning a Boolean Value	
Lab #10		Work on Program #4	
Apr 3	20	6.10 Local and Global Variables; 6.11 Static Local Variables; 6.12 Default Arguments; 6.13 Using Reference Variables as Parameters; 6.14 Overloading Functions (not used); 6.15 The exit Functions (not used); 6.16 Stubs and Drivers *(Begin Oral Presentations on various algorithms)*	
Apr 5	21	7.1 Arrays Hold Multiple Values; 7.2 Accessing Array Elements; 7.3 No Bounds Checking in C++ *(Oral Presentations on various algorithms)*	
Lab #11		Functions with Input and Output Parameters	Lab #11 [10]
Apr 10	22	7.4 Array Initialization; 7.5 Processing Array Contents; 7.6 Using Parallel Arrays; 7.7 Arrays as Function Arguments	
Apr 12	23	7.8 Two-Dimensional Arrays; 7.9 Arrays with Three or More Dimensions; 7.10 A Case Study	Program #5 [50]
Lab #12		Arrays	Lab #12 [10]
Apr 17	24	8.1 Introduction to Search Algorithms; 8.2 A Case Study	
Apr 19	25	8.3 Introduction to Sorting Algorithms; 8.4 A Case Study; Review for Exam #3	
Lab #13		Arrays and Sorting	Lab #13 [10]
Apr 24	26	**Exam #3: Chapters 6, 7, and 8**	
Apr 26	27	10.1 Character Testing; 10.2 Character Case Conversion; 10.3 C-Strings	
Lab #14		Work on Program #5	
May 1	28	Work on Program #5	Program #5 due
May 3	29	Review for Final Exam	
	30	**Final Exam: Monday, May 8th from 11:00 to 1:00**	