California State University, San Marcos General Education Program
GENERAL EDUCATION NEW COURSE CERTIFICATION REQUEST
• AREA B4: Mathematics and Quantitative Reasoning
See GE Handbook for information on each section of this form

ABSTRACT

Course Abbreviation and Number: CS 111	**Course Title: Computer Science I**	
Number of Units:4 _____		
College or Program: CHABSS X CSM CEHHS COBA Other________________________________	**Desired term of implementation:** X Fall Spring Summer Year2014	**Mode of Delivery:** X face to face
Course Proposer (please print):Rocio Guillen	**Email:rguillen@csusm.edu**	**Submission Date:Feb. 2014**

1. **Course Catalog Description:** Emphasizes programming methodology and problem-solving. A high-level language such as C++ will be used for the specification and implementation of algorithms. Includes principles and applications of software engineering, numerical computing, artificial intelligence, databases and user interface. Three hours lecture and three hours laboratory. Students lacking basic computer literacy skills are encouraged to take CS 105 first prior to CS 111. Pre/Corequisite: MATH 160.

2. **GE Syllabus Checklist: The syllabi for all courses certified for GE credit must contain the following:**

	Course description, course title and course number
	Student learning outcomes for General Education Area and student learning objectives specific to your course, linked to how students will meet these objectives through course activities/experiences
	Topics or subjects covered in the course
	Registration conditions
	Specifics relating to how assignments meet the writing requirement
	Tentative course schedule including readings
	Grading components including relative weight of assignments

SIGNATURES

_______________________________ _______________________________

Course Proposer Date Department Chair date

Please note that the department will be required to report assessment data to the GEC annually.

______

DC Initial

Support ☐ Do not support* ☐ Support ☐ Do not support* ☐

_______________________________ _______________________________

Library Faculty Date Impacted Discipline Chair Date

California State University, San Marcos General Education Program
GENERAL EDUCATION NEW COURSE CERTIFICATION REQUEST
• AREA B4: Mathematics and Quantitative Reasoning
See GE Handbook for information on each section of this form

	Support	Do not Support*		Approve	Do not Approve
Impacted Discipline Chair Date	☐	☐	GEC Chair Date	☐	☐

*** If the proposal is not supported, a memo describing the nature of the objection must be provided.**

Course Coordinator: Phone Email:

California State University, San Marcos General Education Program
GENERAL EDUCATION NEW COURSE CERTIFICATION REQUEST
• AREA B4: Mathematics and Quantitative Reasoning
See GE Handbook for information on each section of this form

Part A: B4 Quantitative Reasoning General Education Learning Outcomes (GELOs) related to course content. [Please type responses into the tables.]

<table>
<tr><th>Math/Quant Reasoning GELOs this course will address:</th><th>Course content that addresses each GELO.</th><th>How will these GELOs be assessed?</th></tr>
<tr><td>B4.1: Explain and apply a variety of fundamental computer science and related mathematical concepts, symbols, computations and principles.</td><td>Concepts: Numerical systems used by computers binary, octal, decimal, hexadecimal. How to convert from one numeric system to another using algorithms. Variables and their use in computations. Evaluation of expressions containing arithmetic, relational and logical operators using levels of precedence and associativity rules. Basic constructs to write a program that is syntactically and semantically correct: decision statements, iteration statements, functions. Symbols: flowcharts, arithmetic operators: binary +, unary -, binary -, binary *, binary /, binary %, logical operators: &&, ||, !; relational operators: >, >=, <, <=, ==, !=; assignment operators : =, +=, -=, *=, /=, %=. Computations: programs are written, compiled, debugged and tested for correct application of constructs. Principles: planning a solution for a given problem, describing the solution using pseudocode and/or a flowchart to review the logic of the solution, translating the solution into code following the syntax of a programming language, compiling the code to remove any syntactic errors, testing the program with several test cases to correct logic errors, documenting code for ease of maintenance.</td><td>Students will be given a problem involving one or more of the concepts learned and be expected to solve it using the relevant symbols, computations and principles. Students will be expected to state their solution in a correct, clear and complete manner.</td></tr>
<tr><td>B4.2: Determine which quantitative or symbolic reasoning methods are appropriate for solving a given problem and correctly implement those methods.</td><td>Given a problem plan a solution, describe the solution using pseudocode and/or flowchart. Follow the syntax of the statements in a programming language to write a program.</td><td>The student will be given a problem whose solution will require the implementation of a program. The student will have to make choices about which statements to use and how to put them together for the program to generate the correct results.</td></tr>
</table>

California State University, San Marcos General Education Program
GENERAL EDUCATION NEW COURSE CERTIFICATION REQUEST
• AREA B4: Mathematics and Quantitative Reasoning

See GE Handbook for information on each section of this form

Part B: General Education Learning Outcomes required of all GE courses related to course content:

GE Outcomes required of <u>all</u> Courses	Course content that addresses each GE outcome?	How will these GELOs be assessed?
Students will communicate effectively in writing to various audiences. (writing)	Homework, exam problems and program documentation will require students to explain their thinking.	Students will be expected to write out solutions to problems and implementing these solutions writing programs that must include comments explaining what the different code segments do.
Students will think critically and analytically about an issue, idea or problem. (critical thinking	Problems given will require students to determine what is being asked of them, think about what method/procedure of solution is appropriate, and properly implement that method/procedure.	Students will be expected to solve problems by using accepted principles of computational and mathematical thinking in a logical way

Part C: GE Programmatic Goals: The GE program aligns with CSUSM specific and LEAP Goals. All B4 courses must meet at least one of the LEAP Goals.

GE Programmatic Goals	*Course addresses this LEAP Goal:*
LEAP 1: Knowledge of Human Cultures and the Physical and Natural World.	***No X Yes***
LEAP 2: Intellectual and Practical Skills	***No Yes***
LEAP 3: Personal and Social Responsibility	***No Yes***
LEAP 4: Integrative Learning	***No Yes***
CSUSM Specific Programmatic Goals	**Course content that addresses the following CSUSM goals. Please explain, *if applicable.***
CSUSM 1: Exposure to and critical thinking about issues of diversity.	***X No Yes (please describe):***
CSUSM 2: Exposure to and critical thinking about the interrelatedness of peoples in local, national, and global contexts.	***XNo Yes (please describe):***

Part D: Course requirements to be met by the instructor.

Course Requirements:	How will this requirement be met by the instructor?
Course meets the All-University Writing requirement: A minimum of 2500 words of writing shall be required for 3+ unit courses.	Students will be expected to write out solutions to problems, write program and document the programs.
All courses offered in area B4 must have a prerequisite of at least intermediate algebra and must use a level of mathematics beyond that of intermediate algebra. No remedial algebra courses (e.g., Math 10, 20, and 30) can be used to satisfy this requirement. Even if a course has intermediate algebra as a prerequisite, it will	CS 111 has an explicit co/prerequisite of Math 160, Math 160 makes use of college level algebra heavily in its use of equation solving techniques at and above the level of intermediate algebra. Problems in the course include coding of computational methods in calculus (e.g., calculation of Riemann sums for integrals,

not satisfy the Quantitative Reasoning Requirement unless it also meets each of the following three conditions:	method of bisection or Newton for solving equations. These involve ideas higher than the intermediate algebra. Other notions above the level of intermediate algebra include: conversion between non-decimal number systems, Boolean algebra, and the algebra of other unary, binary and ternary operators introduced in the class.
• It must focus on the use of mathematical language and formal reasoning in a variety of diverse disciplines, using a broad range of examples.	The mathematical language used includes: notions of numerical systems, evaluation of expressions, correct application of arithmetic, logical and relational operators. Formal reasoning includes using the ideas that: in an assignment statement the right-hand side of the assignment operator must be evaluated first before assigning this value to the left-hand side of the assignment operator; functions may or may not have parameters or arguments and may or may not return values; parameter passing by value requires to copy the value and can only return one value; parameter passing by reference only requires the address of a memory cell and can return zero or multiple values; iteration can be count-based or logic-based; the result of a conditional statement is true or false. These ideas are used for each step in an algorithm as a student proceeds to implement a program as the solution to the problem given. The language and reasoning are applied to all disciplines that require of quantitative, consistent solutions. This help students to understand the broad application of the subject. The programming language used is C++, which is suitable for use in a wide variety of areas. Students will learn basic constructs that are used in high-level programming languages. Problems assigned involve applications in the field of physics, biology, chemistry, behavioral sciences, artificial intelligence, among others.
• It must provide some historical perspective on the role which this approach has played in the development of human knowledge and of our understanding of the world.	Students are introduced to the development of computers. First users of computers. The mathematical foundations of Computer Science. The first code, ASCII, used to exchange information. The first programming languages and the emphasis on scientific computation. The new paradigms to implement solutions that are shareable and reusable. How the emphasis on process efficiency moved to data efficiency of usage. Applications that have had a significant impact in how human knowledge is disseminated.
• It must demonstrate a variety of methods, such as the use of abstract symbols, of numeric techniques, of logical reasoning, of geometry, etc.	Methods demonstrated include: usage of variables to represent quantities; usage of rules to evaluate expressions; usage of flowcharts to understand the logic of a solution; usage of a set of principles which are used to justify moving from one statement to the next in a program; usage of functions to show relationships between variables; usage of test cases to verify the logical correctness of a program and correct

	any errors; usage of compilers to verify the syntactical correctness of a program and correct any errors; usage of classes and objects to describe an abstract data type in terms of a range of values and operations on those values. Problems assigned involve applications in the field of physics, biology, chemistry, behavioral sciences, artificial intelligence, among others.
A **statistics component** may be included which must:	
• Develop the students' ability to comprehend the power and broad utility of the fundamental mathematical models presented, rather than merely teaching rote statistical skills; and	N/A to CS111
• Must indicate applications to several areas.	N/A to CS111
A **computer science component** may be included which must:	
• Teach a computer language that is suitable for use in diverse areas;	The programming language used that is suitable for use in a wide variety of areas is C++. Students will learn the basic constructs that are used by high-level programming languages.
• Teach this language in such a way that the student is led to a fundamental understanding of the nature of problem solving by combining data structures with algorithms; and	The approach to teaching a programming language is to start with the most simple concepts and constructs in a programming language to keep building on these concepts and constructs by combining them in more complex structures. The final product is a program that is the code for the steps in an algorithm to solve a problem given a set of specifications.
• Provide fundamental skills in the use of computers for the application of university level quantitative methods to the solution of problems in many diverse areas.	The course includes 3 hours of laboratory where students are required to apply the concepts and methods learned in the lecture by developing solutions to a given problem. The solution is implemented by coding, compiling, debugging, documenting and testing a program written in high-level language such as C++. problems assigned are in many diverse areas.

CS111-10 COMPUTER SCIENCE I
CRN 27191 Spring 14
Class MW 1130-1245 Academic Hall 411A

Instructor: Dr. Rocio Guillén
Office: SCI2 225
Phone: (760)-750-8046
email: rguillen@csusm.edu
Office Hours: MW: 10:00-11:00 AM & M: 13:00-14:00 PM & by appointment

Prerequisite: A **passing grade** (at least a C) on MATH 125 (concurrently enrolled in MATH160/MATH162). Students without proper preparation will be taking this course with a high risk of failling.

Catalog Description: Emphasizes programming methodology and problem-solving. A high-level language such as C++ will be used for the specification and implementation of algorithms. Includes principles and applications of software engineering, numerical computing, artificial intelligence, databases and user interface. Three hours lecture and three hours laboratory.

Textbook Required:
Gary Bronson. *A First Book of C++.* Fourth Edition. Course Technology CENGAGE Learning. 2012.

Course Objectives:

1. To understand the fundamentals of programming.
2. To enhance students' problem-solving skills in Computer Science.
3. To learn how to write computer programs in a programming language, i.e., C++.
4. To learn the basic concepts and principles of a high-level language.
5. Applications of different areas in Computer Science.

Student Learning Outcomes:

1. Application of fundamentals of programming.
2. Ability to use different constructs of a programming language.
3. Knowledge of basic and current applications in Computer Science.
4. Recognition of the importance of enhancing problem solving skills using a programming language.

5. Ability to learn advanced constructs of a programming language based on basic building blocks.

Writing Requirement: The writing requirement will be fullfilled with assignments.

Course Expectations
You are expected to put in **at least eight hours outside of class** for the lecture and similar time for the labs. Reading the material, working on written assignments, writing, compiling and testing programs in the book, and starting work on your programs early will provide you with the knowledge and skills to succeed in the class. The course is 4 credits consisting of the following: 1) a lecture portion during which concepts of Computer Science and programming are taught, and, 2) a laboratory portion during which you apply the concepts learned in class using a programming language.

Attendance and class participation
Your enrollment in this course is contingent upon attendance during the first two weeks **on time**. Absence without prior arrangement with the instructor will be considered as voluntary disenrollment by default. Attendance to both the lecture and the lab is required.

Illness or other verified emergencies.
Written proof is required. The limit to present verified evidence is a week from emergency. Phone calls and email messages do not replace written proof.

Collaboration/Academic Honesty:
Discussing with your classmates general approaches to problems is reasonable and encouraged. However, it is **not reasonable** to copy a solution, or to give away a solution. This will be considered an instance of academic dishonesty, and may result in a course grade of F or another disciplinary action (Check the Academic Honesty Policy in the General Catalog for AY 2012-2014).

Special Needs: Support services are available through the Disabled Student Services office (Craven Hall 4300) for those students who qualify.

Grading:
The grading policy basically rewards two things: **quality** and **timeliness** of your answers to exams, quizzes and assignments given in the lecture. The laboratory portion is graded by the lab instructor and it is 10% of your final grade.

In terms of **quality**: Answers must be correct, results must follow the instructions given for the assignment, and software used must be the same as the ones installed in the computers at CSUSM.

In terms of **timeliness**: Submit assignments by the deadline. No late assignemnts will be given credit. Therefore, I strongly recommend that you do not wait until the last minute to work on your assignment or ask for clarification.

	Points	Points for Final Grade	Max # of Points (Weight)
Exam I	100	pts obtained $\star$ .20	$100 \star .20 = 20$
Exam II	100	pts obtained $\star$ .20	$100 \star .20 = 20$
Comprehensive Final	100	pts obtained $\star$ .25	$100 \star .25 = 25$
Homework	10 each	(tot pts obtained $\star$ .10)/tot pts	tot # homework $\star$ .10 = 10
Quizzes	15 each	(tot pts obtained $\star$ .15)/tot pts	tot # quizzes $\star$ .15 = 15
Labs	100	pts obtained $\star$ .10	$100 \star .10 = 10$
Total			100.00

Final Grade: A:90.0-100.0 B:80.0-89.99 C:70.0-79.99 D:60.0-69.99 F $\leq$ 59.99

Additional Course Requirements and Policies:

1. You are responsible for the following:
   - Computers, laptops, tablets, iPads, and similar devices **<u>must be turned off during</u>** the lecture.
   - All the materials covered in class, homework and labs.
   - Reading announcements posted on the webpage via "Cougar Courses" for the course.
   - Checking your e-mail for any messages related to the course.
   - All iPods, iPads, cellular phones, pagers, and similar devices **<u>must be turned off before</u>** entering the classroom. **No exceptions**. You will be asked to leave the class the first time this occurs. The second time you will be requested to speak with me before attending the next lecture.
   - **No texting** in class. Points will be deducted.
2. **Assignments, Exams and Quizzes**
   - Reading the material before class is an assignment. It will help you to participate actively in class, ask questions about what you do not understand, and answer the questions in the exams.
   - Using the software installed on campus' labs. Programs assigned as homework that do not compile will not be given credit.
   - No make-up exams will be given.
   - Homework will be collected at the beginning of class on the day it is due. Homework will be considered **late after it has been collected, i.e., 11:40 am**. Late homework will not be given credit.
   - No e-mailed homework will be accepted, unless otherwise noted.
   - Discrepancies on grades for homework/quizzes/exams should be submitted in writing within one week from the day that the homework/quizzes/exams are returned back.

- Exams are closed book.
- **There will be a quiz after each chapter/topic is covered.**
- No grades will be emailed/posted.

3. **Software**
   All programs must compile and run on empress using the g++ compiler. Empress' operating system is linux.

4. **Other Resources**
   STEM Center Academic Hall second floor

Tentative Schedule **<u>subject to change</u>**

Week	Topic	Readings in Book
1	Introduction	Chap. 1
1	Overview of Programming	Chap. 1
2	**Quiz 1**	
2	Data Types, Variables & Declarations	Chap. 2
2	Assignment Operators	Chap. 3
2	Interactive Input & Output	Chap. 3
3	Files	Chap. 9
3	**Quiz 2**	
3	Relational Expressions	Chap. 4
3	Logical Expressions	Chap. 4
4	**Quiz 3**	
4	Selection Statements: if-else Statement	Chap. 4
4	Selection Statements: switch Statement	Chap. 4
4	**EXAM I**	Chapters 1-4, 9
5	Repetition (Loop) Statements: while Statement	Chap. 5
5	Repetition (Loop) Statements: do-while Statement	Chap. 5
5	Repetition (Loop) Statements: for Statement	Chap. 5
5	**Quiz 4**	
6-8	Functions Parameter-Passing by Value	Chap. 6
9-10	Functions Parameter-Passing by Reference	Chap. 6
11	**Quiz 5**	
11	Variable Scope	Chap.6
12	**EXAM II**	Chapters 1-6, emphasis on Chapters 5-6
12-13	Arrays	Chap. 7
13	**Quiz 6**	
13-14	Object-based Programming	Chap. 10
14	**Quiz 7**	
14-15	Object-based Programming	Chap. 10
16	**FINAL COMPREHENSIVE EXAM** **Monday May 12, 1:45 - 3:45 PM**	Chapters 1-7, 9, 10, emphasis on Chapters 7 & 10