COURSE NUMBER: CSCI200	**CREDITS:** 3
COURSE TITLE: C++ FOR PROGRAMMERS	**PREREQUISITES:** CSCI120, CSCI125/CSCI127 or equivalent.

Weekly Hours: 3	**Lecture:** 1.5	**Lab:** 1.5	**Total Hours:** 39	**Total Weeks:** 13

COURSE DESCRIPTION:

This course is designed to extend the student’s knowledge of the principles and practice of Object Oriented Programming (OOP) to the C++ programming language. Students must have a thorough understanding of the features of OOP before starting this course. The course begins with a review of basic programming techniques and OOP concepts and progresses to advanced topics in OOP using C++. The examples and exercises require knowledge of fundamental algorithms and programming techniques in an object-oriented context.

TEXTBOOK:

Free online book: Thinking in C++ 2nd Edition by Bruce Eckel:

https://ia802806.us.archive.org/25/items/ThinkingInCVol1/Thinking%20in%20C++%20-%20Vol%201.pdf

LEARNING OUTCOMES:

By successful completion of this course, you should be able to:

Outline the essential features of the C++ programming language.

Describe the fundamental principles of Object Oriented program design.

Describe the basic concepts of Software Engineering.

Define algorithms using pseudocode.

Construct C++ programs from algorithms using standard OOP methodologies.

Apply the concepts of class, method, constructor and object.

Apply the concepts of program data abstraction, function abstraction, inheritance, overriding, overloading and polymorphism.

Describe and apply techniques to debug C++ programs.

Trace the execution of C++ programs.

Use elementary data structures such as arrays and linked lists.

Implement fundamental algorithms such as the linear search and selection sort.

Use C++ standard templates.

Test programs using the methods described in the course.

Document a project.

COURSE CONTENT:

Week	Topic	Chapter
Week 1	Introduction to C++ for Programmers	1
Week 2	Objects	1,2
Week 3	From C to C++	3
Week 4	Data Abstraction	4
Week 5	Hiding the implementation	5
Week 6	Initialization and Cleanup	6
Week 7	Function Overloading & Default Arguments, Inline Functions	7,9
Week 8	Constants, Name Control	8,10
Week 9	References and the Copy Constructor	11
Week 10	Operator Overloading, Dynamic Object Creation	12,13
Week 11	Inheritance	14
Week 12	Polymorphism	15
Week 13	Templates	16

Topics are closely based on material from Thinking in C++, 2nd Edition by Bruce Eckel.

EVALUATION:

Lab Assignments / Participation	15%
Midterm	30%
Final exam	35%
Quizzes / Lab Tests	20%
Total	100%

Midterm Exam – Questions types may include: multiple choice, short answer, interpreting code (predict the output), writing code, as well as other kinds of questions. Questions could be based on the course resources and notes, example programs, and lab activities.

Cheating: Students cheating on tests and exams will receive a “F” grade in this course.

If a student misses an exam, a mark of zero will be assigned unless there are extenuating circumstances. In such cases, the proportion of grade assigned to the missed exam will be added to the proportion assigned to the final exam. The final exam will be held during exam week. NO consideration will be given to any student wishing to write the exam at any other time than that assigned.

It is a student’s responsibility to know and follow the school’s policies regarding cheating on exams.

The school’s policy regarding electronic devices is that any student who has a cell phone or other unauthorized electronic device (ie. ipad, laptop, playbook, etc.) on their person or around their desk during an exam will be guilty of cheating and will a grade of “F” for the course.