****

**C++ Course (Example) Syllabus**

**Instructor**: David Kolesar

**Phone**: xxx.xxx.xxxx

**Email**: david.kolesar@xxxxxx.edu

**Teaching Assistant**: Ned Flanders

**Course Description:**

This course introduces the student to the C++ programming language with an emphasis on enterprise-level development. Specifically, this will focus on giving students hands on experience writing code and solving programming problems. These include making practical design decisions, understanding object-oriented programming, modern design patterns, testing, and using a modern software development environment. Topics covered include flowcharts, problem solving, programming concepts, classes and methods, common data structures. Throughout the semester, students will write their own programs and engage in code reviews.

**Course Philosophy:**

There is a major difference between enterprise level code and academic code. Undergraduates often master theory and successfully matriculate from four year degree programs in computer science, only to later flounder in junior developer positions. While a thorough knowledge of computer science and a powerful command of theory are going to empower developers to make high-level design decisions and give a greater appreciation for the tools they're using, an understanding design patterns, being familiar with practical features of a language, writing code that people know how to read, and following trends of what technologies are used to solve modern programming problems are what's going to get things working at the office.

As both an educator and a software engineer, I wholeheartedly feel that contemporary instructors are faced with recognizing that major changes are warranted with how we teach and prepare the workforce of future software developers to solve problems. That being said, **this course will require the student to actually write code**.

This is where C++ comes in. This is a very special breed of programming language in that it is sometimes considered a mid-level language -- meaning it has elements of both machine languages and high-level languages. Simply put, C++ forces the programmer to manually deal with lower-level processes of the language (such as memory allocation) that would otherwise be handled for them in a higher-level language (such as Java or C#). By requiring attention to such details, it makes an excellent candidate for a theory/practice based course.

My hope is that the student can walk away with from this course with a greater appreciation for both.

**Learning Objectives:**

The goal of this course is to empower the student with a greater command of the C++ programming language. Specifically, the graduate will be able to enter a professional environment with a C++ codebase and be able to successfully contribute by:

* Being able to write enterprise-level code following good design patterns
* Being able to set up a modern C++ development environment
* Being familiar with modern C++ libraries
* Being familiar with common solutions to common C++ programming problems
* Being able to debug programs and identify errors
* Being able to design applications
* Being able to support existing applications
* Having a thorough understanding of the lifecycle of a C++ program
* Understand how to use pointers and smart pointers

**Topics Covered:**

* Introduction to the C++ programming language
* Compiler, linker, and libraries
* The lifecycle of a C++ program
* Setting up your development environment
* Writing basic programs
* Data Types
* Variables
* Operators
* If Statements
* Loops
* Functions(and arguments)
* Scope
* Classes
* Debugging
* Bitwise Operators
* Pointers
* Functions Revisited (Function overloading, inline functions)
* Lambdas (Anonymous Functions)
* Design Patterns
* Object Oriented Programming (Demeter's Law)
* Access modifiers
* Shallow Versus Deep Copying
* Dependencies
* Inheritance
* Polymorphism
* Exception Handling / Validation
* Move Semantics / Pointers
* The Standard Library

**Equality Statement:**

The instructor is dedicated to establishing a learning environment that promotes respect among all people in the classroom. The classroom will operate with respect to all forms of diversity including race, class, culture, religion, gender, sexual identity, and physical ability. It is important that this is a safe classroom environment. We will practice being generous and respectful members of our classroom community. Anyone noticing discriminatory behavior in this class, or who feels discriminated against, should bring it to the attention of the instructor immediately.

**Accommodations:**

Accommodations are collaborative efforts between students, faculty, and the Human Resources (HR). Students with accommodations approved through the DRC are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through HR should contact HR immediately. If you are a student with a documented disability and are registered with HR, please contact me immediately to facilitate arranging academic accommodations.

**Grading Rubric:**

This class is pass/fail. This means that student assignments will be evaluated on a case by case basis and will be awarded a passing or failing grade for the course at the end of the semester.