Kishwaukee College Syllabus
CIS 150 – 5H02
C++ Programming I - Honors
3 Credit Hours, Spring 2018

Course Description

The first course in the C++ language sequence. It emphasizes a disciplined approach to problem solving and algorithm development. Topics will include: input, output, sequence, selection, repetition, functions, arrays, data abstraction, pointers, text manipulation, records, and files. Program design, style, documentation, and testing will be practiced. Programming assignments will be completed outside of class. Three hours lecture/discussion a week. **IAI: CS 911**

Prerequisite: Appropriate Mathematics placement test score or MAT 086 or MAT 098

Meeting Time and Place

Lecture/Lab:	A-1374	
Time:	6:00 P.M. - 8:45 P.M.	Tuesday
Dates:	1/16/18 - 5/16/18	
Withdrawal date:	4/27/18	
MLK Birthday observed:	1/15/18	School closed
Spring break:	3/12/18 - 3/18/18	School closed
Faculty development:	3/29/18	School closed
Good Friday:	3/30/18	School closed
Midterm exam:	3/6/18, 4/10/18	during class
Final exam:	5/15/18	6:00 P.M. - 7:50 P.M.

Instructor Information

Instructor:	David G. Klick
Office:	A-1342
Email:	dklick@kish.edu
Phone:	815/825-9337
Website:	kermit.kish.edu/~dklick/
Backup website:	klickfamily.com/david/school/
Desire2Learn:	https://kish.desire2learn.com/
Division Secretary:	815/825-9380 (Brianna Hooker)
Office hours:	M 1:45 P.M. - 2:30 P.M., 5:00 P.M. - 6:00 P.M.
	T 1:45 P.M. - 2:30 P.M., 5:00 P.M. - 6:00 P.M.
	W 10:00 A.M. - 11:00 A.M.
	R 10:45 A.M. - 11:45 A.M.
	other times by appointment

Expected Learner Outcomes

Upon completion of this course, the student will be able to:

1. get input from the keyboard and place output on the screen
2. use control-flow statements in C++ to achieve branching and repetition
3. define and discuss functions as used in C++
4. code and run programs using library functions and user-written functions
5. define and discuss variable scope and class
6. code and run programs using arrays, pointers and reference variables
7. compare and contrast the basic data types, structures and classes
8. code and run programs including structures and classes
9. code and run programs using data files

Required Text and Materials

1. Vahid Frank and Roman Lysecky. KISHWAUKEECOLLEGECIS150KlickSpring2018 Programming in C++. Zyante Inc. Copyright 2018. This book is an online, interactive book. Sign up at zyBooks.com, enter zyBook code KISHWAUKEECOLLEGECIS150KlickSpring2018 and click Subscribe
2. Internet access
3. A standard modern C++ compiler (available free via the Internet)

Breakdown of Course Requirements

10 programming projects @ 30 points each	300 points
10 labs @ 10 points each	100 points
11 chapter challenges @ 15 points each	165 points
2 midterm exams @ 75 points	150 points
1 final exam @ 85 points	85 points
1 honors project @ 200 points	200 points
--	----------------
Total	1000 points

Final Grade Determination

A = 90 - 100%	900 points or more
B = 80 - 89.9%	800 - 899 points
C = 70 - 79.9%	700 - 799 points
D = 60 - 69.9%	600 - 699 points
F = below 60%	less than 480 points

Grade reports will not be mailed out. Please check KishSOS,

My Student Info, under Academic Profile, Grades, for grade reports.

Course Procedures

1. Students are expected to attend class sessions on time and prepared (Note: CIS 123 class sessions are optional attendance). Students should bring whatever they need to take notes to every class.

2. Food and beverages are not permitted in the classrooms or labs. See a more detailed policy at http://kermit.kish.edu/~dklick/foodDrinkPolicy.html

3. Cellphones, music players, etc. must be turned off in class.

4. Students are expected to spend **time outside of class** completing assignments.

5. A familiarity with computers and the Windows operating system is expected.

6. Depending on the assignment, both digital and hardcopy versions of assignments may be required for submission. The procedure for submitting digital copies of assignments will be explained in class. Make sure you always keep a copy of all of your assignments. The instructor is NOT responsible for network failures, server failures, or student mistakes.

7. The instructor answers many questions via email. Due to the high volume of requests, submissions, and questions received via email, the instructor must prioritize responses. Most questions will be answered (or at least acknowledged) within 48 hours. If you do not get a response when you expect one, please keep in mind that your email may have failed to reach the instructor, or may have automatically been rejected by an email client or server. Please try to contact the instructor again and possibly use the phone or an in-person visit if email is failing.

Make-up Policy

1. Assignments are to be turned in on time. Assignments which are not turned in on time will not be accepted unless individual arrangements are made **in advance** with the instructor. In unusual cases where late assignments are accepted, the cost of being late is ten percent of the total possible points for every portion of a day late, up to a maximum of three days late. For example, an assignment received twenty-five hours past its due date will lose twenty percent of its total possible point value (because it is two days late). Assignments which are received more than three days (seventy-two hours) late will not be accepted and are not worth any points. Exceptions may be made to this rule if the student contacts the instructor before the due date and makes special arrangements **in advance** with the instructor. All late acceptance decisions of this nature are left solely to the discretion of the instructor. This rule does not apply once answers to an assignment have been distributed or posted. Assignments submitted after answers have been released are worth zero points even if the answers are posted one minute past the due date.

2. Answers to assignments may be posted online, handed out in class, or sent via email by the instructor. Once an answer to an assignment has been released, no further submissions for the assignment will be allowed. This rule supersedes all other rules about when late assignments may be accepted. In general, the instructor will try to wait at least forty-eight hours before posting or distributing solutions, but there is no guarantee, so get your assignments in on time.

3. Tests are to be taken at the day and time scheduled. Failure to take a test at the scheduled time may result in a grade of 0 for that test. In the case of an excusable absence or a genuine emergency, the instructor must be contacted as soon as possible, preferably before the scheduled test, to reschedule the makeup of that test in the Learning Skills Center on the day the student returns to campus.

Attendance Policy

Class attendance is strongly encouraged. You are responsible for whatever was covered in class, whether you are there or not. If you must miss a class, it is your responsibility to contact the instructor and make arrangements for notes, handouts, or announcements that were missed. Although attendance is not counted toward the final grade, there may be coursework which is done during class time which may count toward the final grade and may not be able to be taken outside of class time.

Kishwaukee College Policies and Resources

It is the responsibility of the student to be aware of Kishwaukee College Policies & Resources found on this link: kish.edu/kcsyllabuspolicies

Tentative Weekly Schedule

Please note that this schedule and the topics covered are likely to change. Changes will be announced in class. If you are not able to attend class, it is your responsibility to find out what was covered. A more detailed schedule is provided on the course website. Assignment descriptions and due dates will also be posted on the course web site.

Week	Date	Topics
1	1/16	Overview of course and introduction to programming (Chapter 1) **School closed for MLK birthday observance on 1/15** syllabus C++ compilers, MSDNAA downloads intro to zybooks.com writing a simple program, using Visual Studio program structure, basic input and output, comments, errors basic programming concepts
2	1/23	Variables, expressions, and assignment statements (Chapters 1 and 2) identifiers, variables, and constants assignment statements and arithmetic expressions data types in C++

		the binary number system output formatting **Chapter 1 challenge activities due** **Chapter 2 challenge activities due** **In-class lab: Input, output, expressions, calculations**
3	1/30	More variables and basics (Chapter 3) characters and strings overflow number types and unsigned numbers type conversions math functions random numbers debugging style guidelines **Chapter 3 challenge activities due** **Program due: Input, output, calculations** **Honors project: Initial proposal due**
4	2/6	Selection (Chapter 4) the Boolean (bool) data type logical operators relational operators using "if" and "if/else" selection statements using the "switch" selection statement the conditional (?) operator **Chapter 4 challenge activities due** **In-class lab: Selection, calculation, output formatting**
5	2/13	Repetition (Chapter 5) using the "while" statement using the "do/while" statement using the "for" statement nested loops

		increment and decrement operators the "break" and "continue" statements loop counters and sentinel values accumulators **Chapter 5 challenge activities due** **In-class lab: Repetition, input validation** **Program due: Selection, output formatting** **Honors project: User/project requirements due**
6	2/20	Functions (Chapter 6) breaking a program into simpler, modular pieces creating and using simple functions declaring and defining functions calling functions passing values to functions returning values from functions how functions work **In-class lab: Functions, input validation** **Program due: Repetition**
7	2/27	Functions continued (Chapter 6) common errors in functions passing references to functions variable scope and lifetime in functions default parameter values overloading functions unit testing for functions **Chapter 6 challenge activities due** **Program due: Functions, input validation**
8	3/6	Application of concepts so far and Midterm exam In-class demonstration of concepts covered so far **Midterm exam #1**: input, output, variables, calculations, selection, repetition

		Program due: Functions **Honors project: Basic design document due**
	3/13	**School closed 3/12 - 3/18 for Spring Break**
9	3/20	File I/O (Chapter 7), Arrays (Chapter 8) declaring arrays initializing arrays array bounds accessing array values processing arrays **In-class lab: Sequential (text) file input/output**
10	3/27	Arrays (Chapter 8) creating and using arrays of strings passing arrays to functions **Chapter 8 challenge activities due** **In-class lab: Creating and using arrays** **In-class lab: Pointers and arrays** **Program due: Sequential (text) file I/O** **School closed on 3/29 for faculty development** **School closed on 3/30 for Good Friday** **Honors project: Detailed design document due**
11	4/3	Searching and sorting arrays (notes on course website) String and character operations (Chapter 9) char data type operations C++ string access and modification operations C-style strings (char arrays) and associated operations **Chapter 9 challenge activities due** **In-class lab: Sorting** **Program due: Arrays**

12	4/10	Pointers and reference variables (chapter 10)
		declaring pointer variables
		initializing pointer variables
		the address-of operator (&)
		using pointer variables, de-referencing (*)
		dynamic memory allocation
		releasing dynamic memory
		working with pointers
		using reference variables instead of pointers
		types of memory: heap vs. stack
		memory leaks
		functions: passing by value vs. passing by reference
		functions: passing pointers
		functions: passing reference variables
		Chapter 10 challenge activities due
		Midterm exam #2: functions, arrays, sequential (text) file I/O
13	4/17	Enumerations and structured data (Chapter 11)
		defining and accessing structures
		passing structures to functions
		arrays and pointers to structures
		Chapter 11 challenge activities due
		Program due: Pointers, arrays, sorting
		Honors project: Preliminary implementation due
14	4/24	Advanced file operations (notes on course website)
		reading and writing binary file data
		implementing random access files using C++
		In-class lab: Random access (binary) file input/output
15	5/1	Introduction to classes (Chapter 12)
		introduction to objects
		introduction to classes

		defining class members
		defining access: private and public
		the difference between a class and a struct
		constructors and member initialization
		overloading
		destructors
		accessors
		mutators
		Chapter 12 challenge activities due
		In-class lab: Classes and objects
		Program due: Random access (binary) file I/O, structs
16	5/8	Exceptions (Chapter 13)
		exception basics
		using exceptions with functions
		multi-file programs
		separating header and implementation files for classes
		preprocessor directives: include, define
		namespaces
		Program due: Classes, objects
		Honors project: Final implementation due
		Honors project: Reflection report due
17	5/15	**Final exam: 6:00 P.M. - 7:50 P.M., Rm. A-1374**
		comprehensive with emphasis on classes, objects, random access (binary) file I/O

Addendum

Suggested assignment topics:

1. input, calculations, output
2. calculations, selection, formatted output
3. repetition
4. functions, input validation
5. text file processing
6. array processing
7. pointers, array processing

8. functions, implementing a sort of an array
9. binary file input/output
10. classes and objects, multiple file compilation