Kishwaukee College Syllabus
CIS 250 - 5001
C++ Programming II
3 Credit Hours, Spring 2016

I. Course Description

The second course in the C++ language. Abstract data types will be used in the design and implementation of solutions to large-scale problems. Topics include: classes, inheritance, polymorphism, and encapsulation; files and pointers; scope, blocks and dynamic memory; recursion; data structures including stacks, lists, queues, trees; graphs; text processing; and searching and sorting algorithms. Programming assignments will be completed outside of class. IAI CS 912. Three hours lecture/discussion a week.

Prerequisite: CIS 150.

II. Meeting Time and Place

Lecture/Lab:	A-1374	
Time:	6:00 P.M. - 8:45 P.M.	Tuesday
Dates:	1/19/16 - 5/20/16	
MLK observance:	1/18/16	School closed
Spring break:	3/14/16 - 3/20/16	School closed
Good Friday:	3/25/16	School closed
Faculty development:	4/15/16	No classes
Midterm exam:	3/8/16	during class
Final exam:	5/17/16	6:00 P.M. - 7:50 P.M.

III. Instructor Information

Instructor:	David G. Klick
Office:	A-1342
Email:	David.Klick@kishwaukeecollege.edu
Phone:	815/825-2086 x 2320
Website:	kermit.kishwaukeecollege.edu/~dklick/
Backup website:	klickfamily.com/david/school/
Desire2Learn:	https://kish.desire2learn.com/
Dept. Secretary:	815/825-2086 x2030 (Shelley Lawson)
Office hours:	M 1:45 P.M. - 2:45 P.M., 5:00 P.M. - 6:00 P.M.
	T 12:30 P.M. - 1:45 P.M., 5:00 P.M. - 6:00 P.M.
	W 1:45 P.M. - 2:45 P.M.
	R 10:45 A.M. - 11:15 A.M.
	other times by appointment

IV. Expected Learner Outcomes

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

A. generate reports and process text

B. use sequential and random access files to save and retrieve data

C. create, use and destroy objects

D. control access to object member data and member functions

E. overload and override methods

F. overload operators

G. create new classes derived from existing classes

H. declare and use virtual functions

I. implement a number of data types including lists, stacks, queues, and trees

J. discuss graphs

K. discuss algorithm complexity

L. create and manage programs with multiple source files

V. **Required Text and Materials**

A. Malik, D. (2013). *C++ Programming: Program Design Including Data Structures, 6th edition.* Boston, Massachusetts: Course Technology. [ISBN-10: 1-1335-2632-2, ISBN-13: 978-1133526322]
4th, 5th, and 7th editions acceptable also

B. Internet access

VI. **Breakdown of Course Requirements**

10 programs @ 50 points each	500 points
1 midterm exam @ 100 points	100 points
1 final exam @ 100 points	100 points
Total	700 points

VII. **Final Grade Determination**

A = 90 - 100%	630 points or more
B = 80 - 89.9%	560 - 629 points
C = 70 - 79.9%	490 - 559 points
D = 60 - 69.9%	420 - 489 points
F = below 60%	less than 420 points

Grade reports will not be mailed out. Please check KishSOS,
My Student Info, under Academic Profile, Grades, for grade reports.

VIII. **Course Procedures**

A. 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.

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

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

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

E. 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.

F. 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.

IX. **Make-up Policy**

A. 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.

B. 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.

C. 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.

X. **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.

XI. **Class Withdrawal**

A "W" cannot be given as a final grade. The student is responsible for officially withdrawing from the class according to procedures described in the college catalog. Any student that does not officially withdraw from the class will receive a letter grade. The last date for withdrawal for this course can be found at "My Class Schedule" on KishSOS. Kishwaukee College reserves the right to administratively withdraw at midterm those students who are not actively pursuing course objectives or who are in violation of standards of behavior as outlined in the Student Code of Conduct and Discipline. For a copy of the student conduct policy, contact the Vice President of Student Services Office or refer to the Kishwaukee College catalog.

XII. **Incomplete Grade**

All course requirements must be completed by the end date for the course. In the event that extremely difficult circumstances merit granting a student more time to finish course requirements, an "Incomplete" (I) grade may be given. Taking an Incomplete is possible only at the instructor's discretion. To receive an Incomplete, a contract between the student and the instructor must be completed and approved regarding the completion of all remaining work within a strictly defined period of time. If the conditions of the contract are not met, an "I" grade may revert to an "F".

XIII. **Class Cancellations**

Class cancellations due to inclement weather will be announced on local radio stations or posted on the College web site: www.kishwaukeecollege.edu. Students may also call the College at (815) 825-2086. Class cancellations due to instructor absence will be posted on the classroom door and posted at www.kishwaukeecollege.edu/class_cancellations/. Room changes will be announced in advance whenever possible and posted on the classroom door.

Note: Although the text in this section is required by Kishwaukee College, the link to class cancellations no longer works.

XIV. **Academic Dishonesty**

In order to evaluate student work, faculty must be able to trust that the work is original with a student and not the work of someone else. Cheating, falsifying information, forgery, plagiarism, and other dishonest actions will not be tolerated. Sanctions for academic dishonesty are at the discretion of the instructor and subject to appeal as provided in the Student Code of Conduct and Discipline. A complete explanation of the policy and procedures surrounding academic dishonesty are outlined in the Kishwaukee College Catalog.

XV. **Copyright**

As a Kishwaukee College Student, you may have copyrighted materials or software made available to you by the college for course use. Please understand that copyright law may prohibit copying these materials. Violation of copyright laws can lead to prosecution for a criminal offense. You are required to abide by the specific copyright and licensing agreements that apply to each particular piece of software.

XVI. **Recording of Classes/Presentations**

Kishwaukee College prohibits students from electronically recording class lectures and presentations (either by audio, video, picture, or otherwise) unless certain qualifying conditions are met. For more information about this College procedure and the qualifying conditions, please visit: www.kishwaukeecollege.edu/go/recordingclasses.

Note: Although the text in this section is required by Kishwaukee College, the link to the recording policy no longer works.

XVII. **Graduation Requirements for Transfer Degree Students**

All students intending to graduate with an A.A., A.S, A.E.S., or A.F.A. are required to submit a Degree Portfolio. The Degree Portfolio is a way for the institution to measure its effectiveness in preparing students for successful completion of a degree at the transferring institution. Save your college-level work throughout your academic career in order to facilitate the portfolio compilation. Also, submit a completed application for graduation from the college the semester that you plan on graduating prior to submitting the complete Degree Portfolio. Guidelines and the specific requirements (along with the rubrics used for assessment) for the Degree Portfolio can be found at www.kishwaukeecollege.edu/portfolio/.

Note: Although the text in this section is required by Kishwaukee College, the link to the portfolio information no longer works. That information is available through the password protected (it must be important to protect it from the public) school portal.

XVIII. **Assistive Resources Center/Disability Services**

Any student having a documented disability or special learning need and wanting to request accommodations, should contact the Coordinator in A-1317 or at (815) 825-2086 Ext 3960, (815) 825-9106 (TTY), or send an email to awilson@kishwaukeecollege.edu. The student's disability must impact their ability to participate in the educational environment and be documented by an appropriate professional prior to accommodations being approved.

XIX. **Learning Skills Center (A-1300)**

Students must present their Kishwaukee College student ID when using any Learning Skills Center services. For more information on the Learning Skills Center go to www.kishwaukeecollege.edu/go/lsc/. Tutoring is available to all Kishwaukee students free of charge. The Writing Center (Learning Skills Center, A-1306) answers your writing-related questions regardless of the class or assignment. Students are expected to bring all material related to the assignment, including textbooks, paper prompts, etc. Make-up tests, online tests, and placement tests are available through Testing Services, also located in the Learning Skills Center.

XX. **Emergency Procedures/Safety**

Yellow and red Emergency Information flipcharts are located in each classroom. These are quick reference sheets with telephone numbers to reach emergency assistance and a brief description of the correct actions to take in the event of a tornado, fire or other emergency on campus.

XXI. **Religious Observances**

Students faced with schedule conflicts related to a religious observance should make prior arrangements with the instructor a minimum of seven (7) school days in advance of the examination or other activity involved.

XXII. **Student E-mail**

Your Kishwaukee College e-mail account will be the only way to receive official notices from the college. If you choose to forward your e-mail to another account, please be advised that all communication from and within the college will use your Kishwaukee student e-mail. When communicating with instructors or employees of the college, you are required to use your Kishwaukee e-mail address.

XXIII. **Community Resources**

There are numerous community resources that are available to assist students in addressing a variety of personal needs. A listing and description of community resources can be found at www.kishwaukeecollege.edu/go/communityresources.

Note: Although the text in this section is required by Kishwaukee College, the link to the list of community resources no longer works.

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	Reading

1	1/19	I/O and formatting • review syllabus • compiling and running programs on the remote server • assignment submission • using cin, get, and ignore • using peek and putback • detecting input stream failure and using the clear function • formatting output: setprecision, fixed, showpoint, setw, setfill, left, right • additional formatting manipulators: dec, hex, oct, showbase, boolalpha, etc. • I/O using string objects • debugging using cout statements • text file input and output	Syllabus Chapter 3
2	1/26	User-defined functions • function prototypes • void vs. value returning functions • formal paramter lists • parameters vs. arguments • default values for parameters • function overloading • passing by value, passing by reference, and reference variables • "returning" more than one value from a function • variable/identifier scope (global, local) • variable lifetime (static, automatic) • how/where variables are stored in memory: stack vs. heap • testing functions • a brief introduction to recursion	Chapter 6
3	2/2	User-defined simple types, namespaces, string objects • declaring enumerations • declaring variables with an enumeration data type • using enumerations (operators, I/O, passing to/from functions) • the importance of using enumerations • creating and using namespaces (example: kishio I/O library) • declaring and using C++ string objects	Chapter 7
4	2/9	Arrays and C-style strings • declaring one-dimension arrays • accessing a member of a one-dimension array • initializing a one-dimension array during declaration • passing one-dimension arrays to functions (passed by	Chapter 8

		reference) • using a loop to iterate through the elements of a one-dimension array • common errors trying to access non-existent array elements • using an array name as a pointer to the first element • using the const keyword to prevent changes to a passed array • declaring and using C-style strings (arrays of type char) • comparing C-style strings • performing I/O with C-style strings • declaring and using parallel arrays • declaring two-dimension arrays • accessing a member of a two-dimension array • initializing a two-dimension array during declaration • passing two-dimension arrays to functions (passed by reference) • using nested loops to iterate through the elements of a two-dimension array • arrays of objects (such as C++ string objects) • extending array concepts beyond two dimensions	
5	2/16	Structs and classes • defining a struct or class • declaring variables with a struct or class data type • accessing members of a struct or class • specifying public and private access • the differences between a struct and a class • passing structs and classes to and from functions • creating an array of a struct or class type • using assignment with a struct or class • built-in operations • struct/class scope • accessor and mutator functions • constructors • default constructor • destructors • static class members (including initialization of static variables) • the importance of information hiding • UML diagrams of classes	Chapters 9, 10
6	2/23	Inheritance and composition • overriding member functions • constructors of derived and base classes • destructors in a derived class	Chapter 11

		• header files and header guards • protected class members • public vs. private vs. protected • composition	
7	3/1	Exception handling • throwing an exception • using try/catch blocks • rethrowing an exception • creating your own exception class • exception handling techniques • using assertions • exceptions vs. assertions • error handling techniques (terminate, fix and continue, log and continue)	Chapter 14
8	3/8	Pointers, classes, virtual functions, lists, **midterm exam** • declaring and initializing pointer variables • the address-of operator (&) • the dereferencing operator (*) • dynamic memory; using new and delete • operations on pointer variables • creating and using dynamic arrays • shallow vs. deep copies • functions that objects with dynamic memory should implement/override • functions that need special care when using pointers (constructor, copy constructor, assignment operator, destructor) • inheritance, pointers, and virtual functions • abstract classes and pure virtual functions • demonstrate polymorphism • array-based lists • unordered lists • ordered lists • **midterm exam**	Chapter 12
	3/14-3/20	**School closed for Spring break**	
9	3/22	Overloading and templates • the reasons for operator overloading • restrictions on operator overloading • overloading binary operators • overloading unary operators • overloading binary operators • member vs. non-member syntax for overloading funtions • friend functions	Chapter 13

		• overloading the stream insertion operator (<<) • overloading the stream extraction operator (>>) • specifying post-increment and post-decrement operator overloads • overloading the assignment operator • overloading the array index operator ([]) • function templates • class templates	
10	3/29	Recursion • definition of recursion • direct and indirect recursion • avoiding infinite recursion • recursion vs. iteration • when to use (or not use) recursion	Chapter 15
11	4/5	Linked lists • header and implementation files (revisited) • structure of a linked list and its nodes • basic implementation of a linked list • implementing a copy constructor, assignment operator, and destructor • operations on a linked list (insertion, deletion, access elements, display, etc.) • basic introduction to algorithm complexity analysis: linked list operations • templating a linked list • linked list iterators • linked list variation: doubly linked list • linked list variation: unordered list base class • linked list variation: ordered list derived class	Chapter 16
12	4/12	Stacks, queues • structure of a stack (LIFO) • basic implementation of a stack • implementing a copy constructor, assignment operator, and destructor • operations on a stack (push, pop, peek/top, isEmpty/empty) • templating a stack • implementing a stack using a linked list • implementing a stack using an array • stack applications • structure of a queue (FIFO) • basic implementation of a queue • implementing a copy constructor, assignment operator,	Chapter 17

		and destructor • operations on a queue (add, remove, isEmpty/empty) • templating a queue • implementing a queue using a linked list • implementing a queue using an array • queue applications • queue variation: ring buffer • queue variation: double ended queue (deque)	
13	4/19	Searching and sorting • sequential search • binary search • restrictions on binary search (data must be in sorted order) • algorithm complexity analysis of linear and binary search • basic sorting algorithm implementation: insertion sort • basic sorting algorithm implementation: selection sort • basic sorting algorithm implementation: bubble sort • advanced sorting algorithm walk-through: quick sort • advanced sorting algorithm walk-through: merge sort • introduction to binary tree structure and properties • binary tree variation: the heap data structure • minheaps vs. maxheaps • implementing a heap using an array • advanced sorting algorithm walk-through: heap sort • advanced sorting algorithm walk-through: bogosort/Robsort • algorithm complexity analysis of sorting algorithms • sorting arrays vs. linked lists	Chapter 18
14	4/26	Binary trees • properties of a binary tree (revisted) • properties of a binary search tree (BST) • implementation of a binary search tree • operations on a binary search tree: insert, delete, search, traverse, isEmpty • avoiding degenerate binary search trees when inserting sorted data: balanced BSTs • BST traversal: inorder, preorder, postorder • finding a BST's minimum and maximum values • finding the successor or predecessor of a node in a BST • various ways of handling duplicate values in a BST • using recursion vs. iteration when traversing a BST • algorithm complexity analysis for BST operations • BST applications	Chapter 19

15	5/3	Graphs • graph terminology: vertex, edge, neighbor, weighted, directed, acyclic, connected, etc. • graph data structures: vertex list, edge list, adjacency list, adjacency matrix • adding a vertex • adding an edge • breadth-first traversal • depth-first traversal • determining if a path exists • determining if a graph is connected • finding a minimum spanning tree • finding a shortest path • graph applications	Chapter 20
16	5/10	Binary files and random access files • writing binary data • reading binary data • writing to a random access file • reading from a random access file • advantages and disadvantages of binary files • advantages and disadvantages of random access files	Appendix E
Finals	5/17/16	**Final exam: 6:00 P.M. - 7:50 P.M., Rm. A-1374**	