

**Kishwaukee College Syllabus**  
**CIS 260 - 5002**  
**Java Programming II**  
**3 Credit Hours, Spring 2018**

**Course Description**

This is the second course in the Java language sequence. Topics include object-oriented programming, recursion, files and streams, exceptions, string handling, the graphical user interface, searching and sorting algorithms, algorithm complexity, and data structures. Data structures covered will include lists, stacks, queues, trees, and graphs. Three hours lecture/discussion a week. **IAI: CS 912**

Prerequisite: CIS 160.

**Meeting Time and Place**

Lecture/Lab:	A-1374	
Time:	12:30 P.M. - 1:45 P.M.	Tuesday, Thursday
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:	3/8/18	during class
Final exam:	5/10/18	12:00 P.M. - 1:50 P.M.

**Instructor Information**

Instructor:	David G. Klick
Office:	A-1342
Email:	dklick@kish.edu
Phone:	815/825-9337
Website:	<a href="http://kermit.kish.edu/~dklick/">kermit.kish.edu/~dklick/</a>
Backup website:	<a href="http://klickfamily.com/david/school/">klickfamily.com/david/school/</a>
Desire2Learn:	<a href="https://kish.desire2learn.com/">https://kish.desire2learn.com/</a>
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. create classes which use copy constructors
2. create classes which use instance and static variables
3. use access modifiers (public, private, protected, default) appropriately
4. write programs with overloaded and overridden methods
5. declare variables where their scope will be appropriate for the situation
6. use file and stream input/output
7. write code which uses inheritance and polymorphism
8. write basic linked list, stack, queue, and binary tree classes
9. use data structures (linked list, stack, queue, tree, graph) to solve problems
10. create and use abstract methods and classes (including interfaces)
11. apply recursive programming techniques
12. use generics
13. create multi-source file (multi-class) applications
14. implement various search and sort algorithms
15. analyze algorithm complexity and check for correct operation
16. test and debug programs

## Required Text and Materials

1. Internet access (this course posts material on a website and uses Brightspace for assignment submission, discussions, and an online gradebook)
2. Deitel, P. J., & Deitel, H. M. (2012). *Java: How to Program, Ninth edition*. Upper Saddle River, N.J.: Pearson Prentice Hall.

## 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
<hr/>	
Total	700 points

## 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.

## 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](http://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	Reading
1	1/16, 1/18	Classes and objects <b>School closed on 1/15</b> accessing class members compiling using multiple classes/files static vs. instance variables access modifiers (private, public, default) accessor (get) methods mutator (set) methods variable scope, visibility, and lifetime constructors	Syllabus Chapter 3
2	1/23, 1/25	Classes and objects in more depth default constructor overloading constructors throwing exceptions the "this" reference variable public access vs. private access with accessor and mutator methods validity checking using mutator (set) methods	Chapter 8

		<p>predicate methods  composition  destructor (finalize method)  static class members  creating packages  default access (package)  discuss memory concepts of stack and heap</p>	
3	1/30, 2/1	<p>Inheritance  superclasses (base) and subclasses (derived)  objects, inheritance, and software reusability  inheritance hierarchy  protected member access  overriding inherited methods  the @Override annotation  the Object class</p>	Chapter 9
4	2/6, 2/8	<p>Polymorphism  abstract methods and classes  creating and using interfaces  overriding inherited methods  instanceof operator  creating an array of objects  typecasting objects  demonstrating polymorphic behavior  final methods and classes</p>	Chapter 10
5	2/13, 2/15	<p>Exceptions  catching an exception  throwing an exception  the Exception class hierarchy  checked vs. unchecked exceptions  chained exceptions  creating an Exception subclass  assertions  multi-exception catches  try-with-resources</p>	Chapter 11
6	2/20, 2/22	<p>Strings and regular expressions  String constructors  String methods  String immutability  the StringBuilder class  the Character class  tokenizing a String  creating and using basic regular expressions</p>	Chapter 16
7	2/27, 3/1	<p>Files and streams  getting information about a file or directory  reading and writing a sequential access text file/stream  object serialization  catching exceptions during file and stream input/output  using other types of Java streams</p>	Chapter 17

8	3/6, 3/8	Multi-dimensional arrays and <b>midterm exam</b> declaring mutli-dimensional arrays initializing multi-dimensional arrays processing multi-dimensional arrays <b>midterm exam (3/8)</b>	Chapter 7
	3/12, 3/18	<b>School closed for Spring break</b>	
9	3/20, 3/22	Recursion recursion concepts recursion and the method-call stack recursion vs. iteration implementing recursive algorithms	Chapter 18
10	3/27, 3/29	Searching, sorting, and algorithm complexity linear search binary search sorting algorithms (including order $n^2$ and $n * \log n$ ) algorithm complexity using and adapting sorting algorithms <b>No classes 3/29 (Faculty development)</b> <b>No classes 3/30 (Good Friday)</b>	Chapter 19
11	4/3, 4/5	Sorting continued implementing various sorting algorithms deciding which algorithm to use	Chapter 19
12	4/10, 4/12	Generics and collections auto-boxing and auto-unboxing using Java's library classes that implement the Collection interface using an iterator using Java's Collections class using lists, stacks, queues, sets, and maps	Chapter 20
13	4/17, 4/19	Linked lists, stacks, queues using generics creating generic methods and classes implement generic linked-list, stack, and queue classes	Chapters 21, 22
14	4/24, 4/26	Binary trees use a binary tree create a generic binary tree class (delete operation optional)	Chapter 22
15	5/1, 5/3	Graphs discuss graph concepts and terminology walk through creating a vertex list walk through creating an edge list walk through creating an adjacency list walk through creating an adjacency matrix walk through creating a minimal spanning tree walk through finding the shortest path	Not in text

		use and modify a graph class to solve a problem (such as shortest path)	
16	5/8	Graphical user interface (not in text) review GUI topics from CIS 160 draw graphics and text	Not in text
Finals	5/10	<b>Final exam: 12:00 P.M. - 1:50 P.M., Rm. A-1374</b>	

## Addendum

Suggested programming assignment topics:

1. implementing classes
2. implementing classes, inheritance, overloading, overriding
3. implementing polymorphism
4. classes, objects, methods, processing a two dimensional array
5. classes, objects, methods, processing an array
6. modifying and using a linked list with generics
7. modifying and using a linked list and stack
8. modifying and using a binary search tree
9. modifying and using a graph class
10. recursion, graphics