Computer Science 205
Programming II
Fall Semester 2016, August 24 – December 12th
MWF 2:00 – 2:50 PM in CSB 100; M Lab 3:00 - 4:15 PM

Instructor

Dr. Andy Digh
201 D Computer Science Building
Office Phone : 478.301.2816
Cell Phone : 478.747.9395
Email : andy.digh@gmail.com

Office Hours

Monday	12:00 – 2:00 PM; 4:30 – 6:00 PM
Tuesday	4:30 – 6:00 PM
Wednesday	12:00 – 2:00 PM; 4:30 – 6:00 PM
Thursday	4:30 – 7:00 PM
Friday	12:00 – 2:00 PM

Feel free to stop by my office anytime the door is open. We can also set up an appointment if the hours listed above do not fit with your schedule.

I want to encourage you to email me or call me anytime a question or problem arises. Do not ever be afraid to ask any question or discuss any problem whatsoever you may have.

You are welcome to give me a call at home or on my cell anytime, especially if a problem arises which may affect your class attendance or if you need to discuss an urgent issue.

Course Description

A continuation of CSC 204 with an emphasis on advanced object-oriented principles. Topics will include inheritance, polymorphism, graphical user interfaces, event-driven programming, recursion, and simple data structures (lists, stacks, queues, and binary search trees).

Prerequisites

CSC 204 or its equivalent. Mathematical competency is expected.

Course Objectives

After completing this course, students will be able to :

1. Describe the steps in object-oriented software analysis and design that can be used to solve a problem, and be able to specify and implement a user-defined class to represent an abstract data type
2. Create a new Java class from an existing class using inheritance and containment
3. Describe how Java supports polymorphism by using dynamic binding for method calls
4. Perform run-time analysis of simple sorting and searching methods using *Big-O* notation
5. Trace and write a recursive algorithm
6. Use simple data structures such as lists, stacks, queues, and binary search trees
7. Use Abstract Window Toolkit components, including buttons, checkboxes, scrollbars, and text fields
8. Write listener classes by implementing interfaces and extending adapter classes
9. Use threads to implement dynamic applets and collect information using servlets
10. Describe policy approaches in our society today for controlling communication and privacy on the internet

Course Materials

- *Big Java* by Cay Horstmann, Fourth Edition, John Wiley & Sons Publishers, 2009.

- *Ethics for the Information Age* by Mike Quinn, Fifth Edition, Addison-Wesley, 2012. **(Copy to be provided to the student. Chapter 3 will be covered during the last two weeks of the course.)**

You will also need some type of loose leaf ring binder.

Class Attendance and Participation

Attendance is necessary in order to succeed in this class. I will not take roll, but you are responsible for all material presented in class and in the laboratory. If you must miss class, get the notes from a classmate. All class notes and handouts should appear in your notebook.

Students are strongly encouraged to participate in class through discussion and questions. I love lots of questions. There is no such thing as a stupid question when it comes to computers. The only stupid questions are those which are not asked.

Classroom Conduct

Arrive to class on time; if you must be late, please enter and find a seat as quickly and quietly as possible.

Please turn off all cellular phones and pagers before the start of class.

The classroom of a university is a place set aside for instruction and learning. Students are expected to conduct themselves in a way appropriate for adult university students. This will create the best possible classroom environment for learning. Other kinds of behavior, however, which create a less pleasant classroom environment, include working on the computer when the instructor is lecturing, leaving class before it is over, sleeping, reading during class, etc. Please refrain from this type of behavior.

Reasonable Accommodation

Students requiring accommodations or modifications for a disability should inform the instructor at the close of the first class meeting or as soon as possible. The instructor will refer you to the ACCESS and Accommodation Office to document your disability, determine eligibility for accommodations under the ADAAA/Section 504 and to request a *Faculty Accommodation Form.* Disability accommodations or status will not be indicated on academic transcripts.

In order to receive accommodations in a class, students with sensory, learning, psychological, physical or medical disabilities must provide their instructor with a *Faculty Accommodation Form* to sign. Students must return the signed form to the ACCESS Coordinator. **A new form must be requested each semester.** Students with a history of a disability, perceived as having a disability or with a current disability who do not wish to use academic accommodations are also strongly encouraged to register with the ACCESS and Accommodation Office and request a *Faculty Accommodation Form* each semester.

For further information, please contact Carole Burrowbridge, Director and ADA/504 Coordinator, at 301-2778 or visit the ACCESS and Accommodation Office website at
`www.mercer.edu/disabilityservices`.

Grading Policies

I do not know that you have learned anything until you have given me some evidence that you have learned something related to the objectives of this course. This evidence of learning is presented in exams, quizzes, programming and laboratory assignments, and a class portfolio.

This course will consist of 1000 points which can be earned by you in the following manner :

One Hour Exams (3 at 100 points each)	300 Points
Comprehensive Final Exam	200 Points
Quizzes (4 at 50 points each)	200 Points
Programming Assignments (4 at 50 points each)	200 Points
Laboratory Assignments	80 Points
Class Portfolio of Notes and Handouts	20 Points

Grading Scale

Grades in this course will be assigned using a standard ten-point scale based on the total points you have earned :

Points Earned	**% of Points**	**Grade**
900 – 1000	$\geq 90\%$	A
870 – 899	$\geq 87\%$	B+
800 – 869	$\geq 80\%$	B
770 – 799	$\geq 77\%$	C+
700 – 769	$\geq 70\%$	C
600 – 699	$\geq 60\%$	D
Less than 600	$< 60\%$	F

Exams

All exams are required and no makeup exams are given. If you know in advance that you will miss an exam, we can arrange to take it before the scheduled time. In the event that you have an emergency beyond your control preventing you from taking an exam, please notify me no later than twenty-four hours after the exam. If you fail to do so, you will be given a zero for the exam unless you can provide some type of written documentation validating your reason.

If your absence for an exam is excused, your percentage score on the comprehensive final exam will replace your missed exam score.

There were will be three hour-long exams, and a comprehensive final exam. Exams will be a combination of short answer questions, code tracing questions, and code writing questions. You will be writing short code segments on exams rather than complete programs.

One good way to do well on the exams is by simply keeping up with the course on a daily basis and reviewing over your notes the day you took them. Also, since we are learning a language in this course, you need to actually practice the language using a computer. Studying for an exam in a programming course does not involve just studying key terms and concepts. It involves trying those concepts out at the computer, experimenting with them to see what works and does not, and making sure you understand all of the code we've discussed in class and which you have written.

Any student who receives a failing grade on an examination during the course is strongly urged to see me to discuss your work and help you improve in the course.

Quizzes

Four announced quizzes taken directly from the material discussed in class will be administered throughout the semester. Quizzes will consist of several objective as well as short answer questions. No quiz makeups will be given. However, you may certainly take a quiz ahead of time. A missed quiz will be recorded as a zero unless you contact me before the quiz or within twenty-four hours after the quiz. If your absence for a quiz is excused, your percentage score out of 50 on the final exam will replace your missed quiz score.

Quizzes are designed to help prepare you for an upcoming examination by familiarizing you with some of the types of questions you may expect on the material recently covered and allowing you to begin studying the new concepts presented in class.

Laboratory Assignments

Every Thursday we will have an in-class hands-on exercise that we will devote the entire class period toward. The lab will reinforce new concepts and skills recently learned in class, and will help prepare you for examinations. Each lab will be handed out in class by at least the day before lab. **All labs should be completed in your cobra account by 2:00 p.m. on the following Monday**.

Attendance at all in-class labs is mandatory. If you complete a lab prior to the scheduled lab time, you do not have to attend the lab though. However, you must e-mail me prior to lab and you may not continue working on that lab. **You may miss and make up at most two labs. All subsequent labs missed after the second one will be recorded as a zero unless you have written documentation validating your absence.**

Programming Projects

There will be a total of four programming assignments and a total of 200 points for those assignments. This is a substantial portion of your grade, and can improve your overall grade quite a bit. Also, a strong effort on the assignments often translates into a better understanding of the underlying concepts discussed in class and higher scores on examinations. Time spent in trial and error in front of the computer is often where the greatest learning occurs when it comes to computers.

You may complete assignments on any computer on-campus or off-campus that provides communication software as well as a communication link that will allow you to connect to the departmental class server computer (cobra).

Late Assignments

An assignment is defined as late if it is completed anytime after 11:59 PM on the due date. The late penalty is 10% off per weekday. However, the maximum late penalty is 50% off. I will determine your completion time based on the timestamp of the script file in your account.

Program Requirements and Grading

For each program, you will need to create a script file containing the information as specified on the assignment. More information on how to do this will be given in the first programming assignment.

Program output should be reproducible; that is, the instructor should be able to copy your program from your directory, compile the program, run your program and obtain exactly the same output you handed in. You must not remove the program from your directory until after the graded program has been returned to you. You probably should not remove it until the semester is over.

Programs will be graded on the following criteria :

Implementation & Documentation

Correctness Programs should produce correct results for any set of data.

Efficiency & Modularity Programs should exhibit good design principles and careful planning in algorithm development. Programs coded entirely with one main method are not acceptable.

Identifiers All variables and methods should be meaningfully named and documented. No global class variables should be used. The only variables that should be used by a class method are those either declared locally or passed in as a parameter.

Comments & Indentation All meaningful code must be commented. Every method specification should include preconditions and postconditions.

Analysis & Design

A typed description of your problem, specification, and algorithm along with any hierarchy or object-class diagrams should be completed before you begin to code each assignment.

Extra Credit

You will earn extra credit points for handing in an assignment one class meeting prior to its due date. You may also earn extra credit for adding significant additional features beyond what is required.

Program Revision

You may revise any graded program assignment that you do not earn full credit toward in either implementation or documentation. Late programs, programs which do not compile, or programs which do not meet the required specification cannot be revised. You may turn in revised work up to one week after the assignment is returned, and receive back up to half of your points lost. You may only revise an assignment once.

Class Portfolio

All class notes, handouts, assignments, exams, and quizzes should be saved and organized into a portfolio. All handouts, assignments, exams, and quizzes should be saved in some type of a loose leaf ring binder. Notes may be included in the same binder or kept separately in a spiral notebook.

The portfolio will serve as a nice reference you can save for future classes and work involving computers. All topics in the notebook should be legible, arranged in some logical order which is meaningful to you, and easy to find. You should include a table of contents. You do not need to type or re-write your notes unless you so desire. Plastic covers over your work make for a nice extra touch, but are certainly not required.

Cheating

Don't do it. Do not cheat, copy, plagiarize, or represent someone else's work as your own. This is wrong and is a violation of the Mercer Honor Code. Cheating hurts you and your classmates, the course, and the school.

If you suspect cheating by anyone in class, please report it to me as soon as possible. Your identity will be protected and I will act to contain the damage.

Honor Code

Many students have difficulty in determining how to apply the Mercer honor code to computer courses. A few general guidelines should help you in deciding whether you are violating the honor code or not.

1. You are allowed to receive help on your programs from other students, provided the purpose of the help is to help you understand your own program better, not to write your program for you.

2. You are NOT allowed to use copies of programs written by other students, or copies of programs from published sources, even if you plan to modify them extensively.

3. You are NOT allowed to give copies of your programs, or parts of your programs, to other students in any form.

4. YOU MUST WRITE YOUR OWN CODE. DO NOT COPY PROGRAMS OR PARTS OF PROGRAMS FROM ANY SOURCE UNLESS I TELL YOU TO DO SO.

Any violation of the above policies will be treated as academic dishonesty and a violation of the Mercer Honor Code.