CS/IS 135 – Programming in C/C++
Spring 2017
SR 313 (Ticket #1240)
Mondays 6:55 – 10:05 pm

Instructor: Kristina Shroyer kshroyer@glendale.edu

Office Hours: **Location:** SG 154

Mondays: 4:40 – 5:10 p.m. **
Tuesdays: 4:50 - 6:50 p.m. **
Wednesdays: 4:40 – 5:10 p.m. **
Thursdays: 4:50 - 6:50 p.m. **

****IMPORTANT NOTE REGARDING OFFICE HOURS:** I do have classes Monday and Wednesday that start at 5:10pm. That means if you show up at office hours at 5:05 or 5:00pm to start big questions/problems on those days we won't have enough time to do anything. In addition, on exam days (there aren't too many) I will have to end office hours a little earlier in order to make sure the student exams for classes those nights are set up. On those days I will email you on Moodle and let you know that the office hours will end a bit early. On those days if you have a quick last minute question you can come to the class where my test is to quickly ask BEFORE the exam starts – once the exam starts no questions though!

Office Hours: These office hours will be held in person in my office SG 154.

Questions: **Questions may be asked during office hours, during break, by email, or on the question forum in Moodle. The best place to ask is on the question forum in Moodle because that way other students with the same question can see the answer and students can also help each other by answering other students' questions.**

Please do not contact me by phone. I am only in my office during office hours and I don't check my voicemail, email rather than call.

MOODLE QUESTION FORUM: On the first day of class I will show you the Moodle question forum. You can ask any and all questions here (only general class questions save any personal questions for email). Asking here is best because then other students with your same question can see my answer. Also students can help other students here – if you know the answer to another student's question, do answer it!!☺

EMAIL: You can email me questions if the forum doesn't work or you can't get to office hours. I will answer emails as soon as I possibly can. However, I unfortunately ***can't*** be on the email 24 hours per day, seven days a week**.** I do work days at another job so emails sent during those times will likely be returned either that night or the next morning. **Plan ahead, asking for homework help the day it's due or the night before is not a good strategy and it is NOT ok to expect a response at the last minute. I can NOT guarantee emails sent that last minute will be returned in time.**

I will make sure and check email once a day Monday-Thursday and on Sundays. I can not guarantee I will be able to check and respond to email on the weekends (meaning Friday and Saturday). That means you want to get your questions in Sunday-Thursday to get a faster response. **Make sure and put the course name in the subject line of the email and the word "Question" (Example: Question: CSIS 135).**

Please email questions directly to kshroyer@glendale.edu if possible. You may ask on Moodle but realize any reply I give will go to your GCC EMAIL – so if you don't activate your GCC email you won't get my reply…activate now if you haven't yet.

I. Course Description:
CS/IS 135 is a course in programming using the C++ language which is easily transportable with uses in many different applications including but not limited to game programming, business applications, image processing, and systems programming. Types, operators, control flow functions, object-oriented programming, classes, data abstraction, pointers, and arrays are some of the topics covered in the course.

Prerequisites: **CSIS 112 or equivalent (THIS PREREQUISITE IS REQUIRED – no exceptions). In order to pass this class and do well you have learned the basic programming skills in some programming language.** You should understand basic program structure and programming concepts and be able to apply those quickly as we will learn the basics of the C++ language very quickly. **I will NOT be re-teaching basics like pseudocode, variables, what a program is, data types, loops etc – I will quickly review these topics and point out differences between C++ and Java – you should KNOW these concepts from Java (CSIS 112 class) or your EQUIVALENT programming class from another college. I will provide detailed slides that go over these topics though.**

****IMPORTANT*****
THIS IS NOT AN ENTRY LEVEL CLASS. **No programming experience or a few youtube videos is NOT ok for this class.** You should at least know the basics regarding pseudocode, data types, selection statements and loops. I will not be covering these in detail; instead we will do it in a review like style emphasizing the differences between C++ and Java. We will go through these basics very quickly and some that is exactly the same I will only spend a few minutes on. We will have assignments on the basics as well as in class exercises/quizzes.

You are EXPECTED to know how to use a computer and install software – not being able to install software is NOT an excuse for late assignments or doing poorly on exams. I will give you several options and as a college student in a NON entry level course you should be responsible to get everything working – start doing this on the first day of class. These are basics included in the REQUIRED prerequisite.

A lot of people in previous semesters took this class with no programming experience – the first programming exam is a review type exam (basically everything we did in CSIS 112 in C++ with a few little additions) and they barely got through that exam and thought they could complete the course – this was NOT the case. What happened was after our review the course material gets fairly difficult and most of the people with no programming experience struggled to get a C in this class, some were not able to get the C. There is always an exception to the rule, but just remember **most people are NOT the exception** – I did have one or two students with no programming experience get an A, but that was the result of hours and hours of extra studying and dedication to this class. I had a lot of people with no programming experience copy homework assignments and then fail exams. ***I STRONGLY recommend that if you do not have any programming experience at all you take CSIS 112 or some other entry level programming course (it could be in any language) before taking this class. If you are unsure if your programming experience is enough most likely you will struggle in this class.*** I don't want people to get bad grades – especially if the reason isn't that they aren't working hard but more that they don't have the recommended preparation.

In addition, I am NOT the tutor for students without the prerequisite – if you start asking questions about things you should know from the recommended prerequisite I will nicely suggest finding a tutor or perhaps taking CSIS 112 instead of this class. It is not ok to expect me to be your free tutor because you do not have the prerequisite.

IF YOU IGNORE THIS AND TAKE THIS COURSE WITHOUT THE PRE_REQUISITE READ THE FOLLOWING:

- **IT IS EXTREMELY LIKELY YOU WILL GET A C THIS CLASS (sometimes lower) WITHOUT THE PREREQUISITE**
  - **Some schools like UCLA will deny your transfer if you get a C or a W in this class. I can NOT give you special treatment and give you an A or B when you get low scores in the class – it isn't fair to other students**
  - **Check with your counselor if you are still thinking of taking this class without the prerequisite**

ADDITIONAL NOTE: If you learned C++ from an OLD book or from a teacher who teaches 1970's, 80's or 90's style code you will need to RE-LEARN the newer style C++ (the old stuff is MORE THAN two standards gone now). If you don't relearn the newer style of C++ (second standard from the 1990s or higher) you will not do well on exams (so no pre-1990 C++). In this class the second standard will get you most of the way through but the FIRST standard (from BEFORE 1990) will not be acceptable. We will learn a few things from the newest standard as well but most of what we learn in this class hasn't changed that much in the new standard – the new standard is covered much more in Advanced C++ (CSIS 137).

II. Course Plan and Objectives:

Course Aims/Objectives

At the completion of this course you should have a good understanding how to code and implement intermediate level programs in the C++ language as well as an understanding of the programming concepts used in the language. You should also understand that the basic programming concepts you learned can be applied to all programming languages. By understanding the programming concepts presented in this course you should be able to apply those concepts to quickly learn and understand other languages. In the second half of this course we will focus on built-in arrays, pointers, and object oriented programming and you should be able to program an intermediate level object oriented program by the end of the course. I will also introduce you to the new C++ standard Array data type (different from built-in arrays) near the end of the course. In addition, you will be expected to have a basic knowledge of how to use the Microsoft Visual C++ IDE and/or X-Code by the end of this course.

Uses for what you learn in this course

In this class we actually do programming, we don't just talk about programming. Programming is a marketable skill – **most (if not ALL) companies hiring programmers now give tests (so an 'A' in a class will NOT cut it) during or after the interview** – you need to know how to program to get a job – there will be no way to cheat your way out of the test the employer gives. Computer Software and Computer Hardware Engineers also need to know how to program. Science majors will often get involved in research projects and have to read code and modify it for projects they are involved in. I had a geology major in my C++ class a once that ended up needing programming skills for an internship studying volcanoes in Hawaii – the C++ class got her the internship. And math majors often become programmers – the point is this is a marketable skill that could make you money in the future – it may be hard at first but once you catch on it gets much easier – similar to learning a new language.

Recommendation for Serious Programmers:

Take CSIS 137 – Advanced C++, CSIS 139 or the Python Class (CSIS 151). The reason is you want to know objected oriented programming at an intermediate/advanced level to be ready for the real world. This is necessary for game programming, GUI programming, and is necessary for any big project type of programming. This is also essential for software engineers that want to design software projects. In this class we just start on object oriented programming, you need the advanced concepts taught in the advanced classes as well.

Course Topics (General List – not all inclusive):

-Introduction to C++
    -The Compiler
    -Basic Program Structure
    -Basic Input Output
    -Arithmetic Operations
    -Variables and Data Types
-Selection (Relational Expressions, if/else, switch)
-Repetition/Loops (while, for, do while etc.)
-Functions (Including Pass by Reference function arguments)
-The String Class
-Character Manipulation
-Built-in Arrays
-Pointers
-Strings as Character Arrays
-Structures
-Classes/Object Oriented Programming
    -Concepts and basic programs

-The new C++ Array data type (NOT the same as built in arrays)
-Introduction to the this pointer and inheritance

III. Format and Procedures:

Each week of class will consist of lecture and class exercise you will do at the conclusion of class or a quiz you will take during lecture that will be due at the end of class. Do not panic about the class exercises or quizzes. They are designed to make sure you are paying attention in class (so the purpose isn't really "grading" it's learning): the in class exercises will be graded on reasonable effort only and will likely be a programming exercise you can do yourself or with a group; the in class quizzes will be based on the reading for that class and the lecture I gave on that reading, they will be open book, notes and you can discuss the answers with other students.

A. Participation/In Class Exercise/In Class Quiz Grade

In the past I have had students sign in for attendance/participation, this hasn't helped students because they just came to sign the attendance/participation sheet rather than to listen to lecture (which was the point!!). So we will have a different type of participation grade. Due by the end of each class, you will have either an in class exercise (some sort of programming exercise), an in class quiz (true/false and multiple choice), or some days I may just bring a sign in sheet (that will be rare most likely it will be a quiz or exercise). Most of the time you will be taking an in class quiz (there will likely be one each week). The quiz will open at the beginning of class (so you can answer questions as I lecture) and due at the end of class (it's participation for the day). The quiz is open everything (including neighbor), unlimited attempts. The idea is for you to listen and participate so it's not a quiz as much as it is an in class participation exercise.

DO NOT PANIC. These are not designed to be hard. For the in class exercises, points will be given on reasonable effort only (not on perfect programming exercises) and you will be allowed to work in groups if you want. HOWEVER, EVERYONE SHOULD SUBMIT A COPY OF THE EXERCISE ON MOODLE EVEN IF YOU WORKED IN A GROUP. For the quizzes, I will give you plenty of time, unlimited attempts, and they are open book and you may discuss answers with other students – the point is just to make sure you listen in lecture or read the lecture slides, so these are effort based points, take the quizzes over and over until you get 100%!

THERE WILL BE NO MAKE UPS ON PARTICIPATION/IN CLASS EXERCISES/IN CLASS QUIZZES. There are NO EXCEPTIONS to this rule. REMEMBER: this is only worth 2% of your grade, so missing one quiz will NOT affect your grade, do NOT panic. HOWEVER - Do not email if you miss class, there are NO MAKE UPS – NO EXCEPTIONS.

B. Homework/Programming Assignments/Software and Submission of Assignments in Hypergrade

- **SOFTWARE NEEDED FOR THIS CLASS**

  For this class at school we will be using the professional version of Visual Studio that is already installed on the class computers (you will need to get a compatible software at home – see below for my recommended FREE software to get at home)

  - ***IMPORTANT!!*** – DO ***NOT*** GET VISUAL STUDIO **"CODE" (note the word "code" instead of "community")** – it is quirky to set up and doesn't work the same as what we are doing in class so using it would be more difficult.

  - **FOR PC USERS: you should get the free version of Visual Studio Community 2013 – I DO NOT RECOMMEND GETTING 2015 it was buggy for me – get – USE THIS LINK – do NOT GOOGLE you will get confused and get the wrong thing.**
    https://www.visualstudio.com/en-us/news/releasenotes/vs2013-community-vs

  - **FOR MACINTOSH USERS: I will be SUPPORTING the use of X-Code for your homework, XCode is a free Mac App for C++ development.**
    - **What this means if you are a Mac user is you DO NEED to know Visual Studio Commmunity 2013 for class (you need to know it for exams) BUT for homework I want you to use X-Code:**

- **IMPORTANT NOTE FOR PC USERS: I highly recommend using the link I provided and installing Visual Studio 2013 – I do NOT recommend getting 2015 (it has been buggy) and DEFINITELY do not get 2017! Plus 2013 will be more compatible with our school version**

- **If you have a PC here is the link to download Visual Studio Community 2013 (MAKE SURE TO USE THIS LINK TO GET THE CORRECT SOFTWARE):**

    - **MAKE SURE AND UNINSTALL ANY PREVIOUS VERSIONS OF VISUAL STUDIO BEFORE DOWNLOADING!!! – THIS IS VERY IMPORTANT!!!**
      https://www.visualstudio.com/en-us/news/releasenotes/vs2013-community-vs

- **If you have a Macintosh – HOW TO GET X-CODE**
    - It's easy! Go to App Store – Search for XCode and download the free app – it's a little blue box with a hammer on it
    - I do have a Mac and as long as you have XCode I can help you – however that is the only development tool I support in this class for mac
    - **X-Code Information**: https://developer.apple.com/xcode/

- **IMPORTANT FOR MAC USERS!!: Remember just because you are using XCode at home you STILL need to know Visual studio professional (what we have on the school computers) – the reason is I do NOT allow laptops for exams and you will have to use a school computer with Visual studio professional to take your exam.**

- **IMPORTANT!!!! Make sure you start working on getting yourself set up at home ASAP – this way if problems arise we can get them worked out and you won't get behind in class. If you can't get set up in time you'll want to either do the first homework at school or find a computer that works – not being set up is not an excuse. We're going to talk about this on the first day of class in detail once we start lecture. The first homework is easy (assuming you have CSIS 112 or equivalent) so even if you have install issues I'm betting you can get this done at school easily if you have the required preparation required for the course. Just plan in advance and know if this is an issue for you.**

NOTE: This is another reason you need programming experience for this class – this is not as easy as using Java/JGrasp and you don't want to be struggling with the basics as you get your computer all set up at home.

******SUBMITTING /TURNING IN PROGRAMMING ASSIGNMENTS:**
We will be using the Hypergrade online grading system to submit ALL homework (programming assignments and non-programming assignments – I'm fairly sure in this class I made all the assignments programming assignments). Hypergrade is the only thing I require that you purchase for this class; the cost is only $10. **The ONLY way to submit homework will be using Hypergrade – no homework will be accepted via email (no exceptions).** We will go over the details of getting and using Hypergrade in class before your first assignment is due. Make sure to follow instructions when completing assignments – there is a reason for the way each assignment is asked for, make sure and submit the proper files in the format asked for.

You're going to see that projects in Visual STud and XCode are large. However, all that you will need to turn in on Hypergrade is your .cpp file(s) and later on your .h files. I will show you how to locate the files you need to turn in on Hypergrade.

During the first week of class I will go over Visual Studio Community 2013 (don't worry that we have the professional version at school the two versions are so similar you only need to be able to use one to know both) and where/what the files in question are. Do not worry, I will show you how to use this program and install it at home and how to submit assignments – all you have to do is pay attention in class and listen.

C. Midterm Exam Results Can NOT Leave the Classroom
When I hand back the graded Midterm Exams you will have time to look at them during the last 15-20 minutes of class. You cannot take the test with you – doing so will result in a zero on the exam. YOU CANNOT TAKE A PICTURE (this is NOT high school and the point of not letting you take the exam is OBVIOUS). If you take a picture of the exam I WILL take a picture of you doing so and report it to the Dean at the school and let her take it

from there – if you're here to cheat please take another instructor. If you would like more time to look at your test results/errors I will have the tests in my office and you can come look at them at any time during office hours. If you can't make office hours we can schedule an appointment.

D. Important Dates (double check the GCC website for administrative dates which came from there)
(NOTE: Test Dates are Tentative and could change based on the classes progress)

03/04/17	Deadline to add a first 16 week class Last day to drop without a 'W' and be eligible for a refund
03/10/17	Last Day to Apply for Pass/No Pass
04/03/17	Objective Midterm #1 (SECOND HALF OF CLASS (**YOU NEED A SCANTRON**) – tentative date– may change)
04/10/17	Programming Midterm #1 (SECOND HALF OF CLASS - tentative date – may change)
04/17-04/23/17	Spring Break – NO CLASS
05/20/17	Last Day to Withdrawal with a 'W' (double check this date with admissions and records – sometimes the online calendar is incorrect)
06/05/17	Objective Midterm #2 (FIRST HALF OF CLASS – **YOU NEED A SCANTRON**)
06/12/17	**Programming Final 7:50pm-10:10 pm (NOTE THE START AND END TIME)**

IV. Class Notes/Hypergrade/Moodle/Textbook and Supplies:

Class Notes: I will be providing a set of slides/notes for this class. These are DETAILED NOTES that are ***my original work*** that I created to help students, they are not "slides" in the traditional sense nor are they "notes" to assist me in my lecture. I don't need them for lecture instead I created them so I would have documentation of lecture that is very detailed that you could go back and read later. I use them to guide the lecture and to provide details of my lecture you can read later.

We will use Moodle (an online class system) for this course – ALL COURSE INFORMATION: slides, examples, and programming assignments will be posted for you on Moodle. I usually get the slides up for a class on Moodle the night before or morning of class. ***Remember all homework must be completed in Hypergrade; this is true even though the directions for the assignment itself are also going to be posted in on Moodle.***

Hypergrade (REQUIRED for this course – NO EXCEPTIONS): This is the online homework system you will use to submit ALL homework assignments. It is the only thing I require you purchase for this course, the cost is only $10. You need to register for Hypergrade ASAP – I will be providing detailed instructions on how to register for Hypergrade so you can do that within the next one or two days.

Moodle (REQUIRED – you must sign up, it is FREE I'm going help everyone get in the first day of class if they haven't already (http://moodle.glendale.edu/), everyone registered for this class is already registered in Moodle and we will all log into Moodle on the first day of class so I know everyone got in at least once. Moodle will have ALL example programs, lecture slides (in different formats), and other class information in it. **It is REQUIRED that everyone get signed into Moodle** – I will post important announcements and all course information there, it is your responsibility to check Moodle and have up to date information, if you don't sign into Moodle and don't know what is going on that is NOT an excuse for missing an assignment or exam**. In addition grade information will be posted in Moodle, I will NOT (it is not allowed) give grade information via email (it is against school policy).**

Scranton Sheets (REQUIRED for Objective Exams – NO EXCEPTIONS): It is your responsibility to bring a Scranton to both objective exams. NOTE: you can buy them from the bookstore or administration BUT I believe they close at 7 or 7:30 so get them EARLY. There also may be a vending machine in the San Gabriel building but those don't always work so don't' count on them. No Scranton means no exam and that exam will be the one you end up having to drop from your score – don't let this be you.

Software: You are required to get the free software at home (will be explained in class) – if you have a PC you will get Visual Studio Community 2013 – Desktop Version and if you have a Mac you will get XCode.

Text: *First Book of C++*, by Bronson, Fourth Ed.

Publisher: Thompson Course Technology
ISBN: 978-1-111-53100-3

This text is the one I model the slides after but **you may use any current C++ textbook that uses the ISO Standard (no OLD books). I'm also ok with anyone using the Third Edition of this book.

Recommended (HIGHLY RECOMMENDED) Supplies: All students are STRONGLY encouraged to bring a USB flash drive to class to save class examples, your programming assignments, and notes on. Since we may have some time to work in class you need a way to transport your C++ projects between school and home. You want to transport projects rather than C++ files and they are rather large.

V. Grading:

(a) Programming Assignments	5%
(b) Participation/In Class Exercises & Quizzes	2%
(c) **Two best** of the following three exams: Objective Midterm #1 Programming Midterm Objective Midterm #2	62% (so 31% each for your best two exams)
(d) Programming Final	31%

Grading Scale (exams are curved at instructor discretion based on results)

90% and above	A
80%-89%	B
70%-79%	C
60%-69%	D

Programming Assignments: All homework assignments (programming and non-programming) will be done in Hypergrade (an online homework system). ***No emailed assignments will be accepted, it is REQUIRED you submit assignments in Hypergrade to get a score – emailed assignments receive zero points.*** Hypergrade will automatically compile and test code assignments so you will have an idea of whether or not you did the program correctly before I grade it for reasonable effort. ***IMPORTANT NOTE:*** Hypergrade can NOT tell whether you followed instructions and techniques from class, so I will give the final score (**at the END of the semester based on reasonable effort**), Hypergrade will not. In addition, do not worry about spelling mismatches on Hypergrade or small spacing errors – HOWEVER, all calculations must match Hypergrade. Last semester people submitted things that didn't even run in Hypergrade and claimed it was a spacing error!! That won't work. However, if you follow instructions from class and Hypergrade says your code compiled and the test cases ran correctly (all calculations and main formatting must match, only spacing and spelling can be off) you will have a good idea you did well on the assignment.

It is very important to complete the programming assignments. . Programming tests will be based on the assignments given so completing programs successfully will lead to success on the exams. Note programming assignments are worth 5% of your grade – ***you must pass the tests to pass the class***, programming assignments are there to assist you in passing. Programming assignments are graded somewhat leniently (on REASONABLE effort only – since the goal is to learn; **if you do the programming assignments, they compile and run the tests correctly in Hypergrade (excluding spelling and spacing errors), and they have met the required specifications, you should easily get the 5% or close to it. Programming *tests* however are graded in detail, looking for specific concepts.** ***Do NOT rely on the programming assignment grade to pass the class though; you MUST also pass the exams. You will fail the exams if you copy someone else's homework (it happens all the time) – it's up to you to learn and do your own work.***

******PROGRAMMING ASSIGNMENTS ARE GRADED ON EFFORT ONLY AND SOLUTIONS WILL NOT BE EMAILED BECAUSE EXAMS OFTEN HAVE CODE VERY SIMILAR TO**

THESE ASSIGNMENTS – so if I emailed the answer I would be giving you the test!!! Not fair. I will HELP you in office hours to get the assignments correct, but I can't email answers for ANY REASON – NO EXCEPTIONS. If you copy someone else's assignment that's your issue because on the exam you won't know what to do, so you will pay for it on the exam.

Programming Assignment Grading/Feedback: **I do NOT put "points" on any of the homework assignments until the LAST WEEK of the semester – the reason is the grade is based on reasonable effort only and you are only doing the homework to prepare for exams.** There are two reasons for this: the first is the tests often have some code VERY SIMILAR to homework so if I gave you exact answers I would give away answers. The second reason is unfortunately there is just not enough time in the semester. DO NOT WORRY THOUGH. **If you want more feedback you can ask me during the last 15 minutes of class (when we have time), or come to office hours. I will give you help and tips to guide you into getting good code in preparation for the exams.**

Late Policy for Programming/Homework Assignments: Hypergrade does NOT allow late homework submissions. However, Hypergrade does allow me to provide you with "Late Days". I give you 10 "Late Days" to use during the semester. When you see an assignment in Hypergrade there will be a button under that assignment that allows you to use 1 late day and add 24 hours to the due date for that assignment. You can use multiple late days on the same assignment but once you've used all your late days there won't be any more.

Best two of three exam policy on "Midterm" Exams: You will have three exams other than the final – for lack of a better term I'm referring to these as MIDTERM exams. The three midterm exams you will take are: the programming midterm, the objective midterm #1 and the objective midterm #2. You get to drop the LOWEST score of these three exams, in other words you keep the best two of three. If you miss an exam for ANY Reason (it doesn't matter what the excuse is) THAT will be the exam you drop the score of (since it will be the lowest score). There are NO exceptions, things happen I understand and that is why you get to drop the lowest score of your three midterm exams.

Exams (Objective and Programming): may or may not be curved depending upon how the class performs. For programming exams I will have a specific set of code techniques I am looking for.

- Programming Exams are open book/open programming assignment source code (only your source code from your assignments will be allowed as notes)/open computer/closed Internet/closed communication devices.
    - NO FLASH DRIVES WILL BE ALLOWED AND NO SOURCE CODE FROM STUDENTS FROM PRIOR SEMESTER CLASSES ALLOWED. I know which students have taken past exams and given them out, I will check your desktop after the exam and if those students' exams are on it you will not pass (the "trash" on your desktop will be locked for the exam). It doesn't matter that the test has changed, cheating is cheating.
- Objective Exams are closed everything except for TWO sides of ONE handwritten note sheet (I will explain the specifics of this as we get closer to the first test). These exams will be partially Scranton (multiple choice and true false) but WILL ASLO have a portion of the test that will require you to read code and write the code output (this will become more clear as we go – if you took CSIS 112 you know what this means). I will give you some sort of review guide before each objective exams – the general idea is that you should know the information from the slides (lecture notes). Your note sheet may NOT be anywhere except right in front of you on your desk, if it is on the desk in between you and your neighbor you won't be allowed to use it.

Exam Make-Up Policy: **There will be NO MAKE UP EXAMS ON MIDTERMS (the first three exams) in this class given for ANY REASON.**

The reason I can have this policy is you have three midterm exams and keep

only the highest two scores. What that means is if you miss an exam that can just be the exam score you throw out

****There will be NO MAKE UP EXAMS given for the FINAL for ANY REASON. Finals are almost impossible to make up in a class where you need to use a computer during the exam. Check the final date now, if you can't attend you should take this class during a semester where you can attend the exams.**

Participation/In Class Exercises/In Class Quizzes: There will be NO MAKE UPS on theses for ANY reason. It's basically a participation grade – missing one or two will NOT effect your final grade since this is only 2%.

What happens if you are on the "edge" of a grade at the end of the class? For example at 89%: I often will bump these grades up especially if you are a student that attends class, doesn't cheat (just because I didn't say anything to you doesn't mean I don't know if you cheated), has a good attitude, and doesn't TALK during class (yes I notice and yes I make notes in my personal file – no I don't tell you I made the note). In other words come to class and don't talk and don't cheat and at the end I will help your grade the best I can!! ☺

VI. First Day Drop Policy

This is a full class. Anyone who does not show up for the first class (without emailing me **in advance**) will be dropped from the class to allow students trying to add into the class – this includes students on the wait list. That means we should know by the second class how many people are going to be able to add the class. I will add people in the order they are on the wait list. At the beginning of the second class we will take care of adds based on drops that have been processed. **If you don't attend class and I am adding people you will lose your wait list position and I will add the next person on the list – come to class to get added.**

VII. Recording and Photographing policy: I do ***NOT*** allow pictures of me to be taken in class for ANY reason. There are NO exceptions to this rule and I am very serious about it. If you want to record lecture the policy is you are required to ASK me first and I will likely agree provided you sign something saying the recording will NEVER be published anywhere for any reason (especially the interenet).

VIII. IMPORTANT DROP POLICY (after the first week of class you MUST drop yourself – know the drop deadline)

If you decide to drop the class, you can't simply stop attending or just not go to exams and expect me to drop for you. I'm not supposed to do that. That means it's your responsibility to withdrawal if you need to – make sure and do it, you don't want to hurt your GPA by getting a failing grade when you could have dropped or gotten a W.

IX. Class Rules

-NO TALKING WHILE I AM LECTURING (PLEASE!! You have Chat and Note Passing if you absolutely need to communicate) – talking distracts and disturbs the other students and is rude/not cool. This includes loud whispering. First violation will be a warning. If you have continued violations it could result in a loss of your attendance points for the day**.** This is a common violation in this class, please respect the class and the other students by not talking or whispering during the class, whispering may seem quiet but in a small room like this ends up being louder than you think. ***When it comes to the end of the class and you have an 88% I will remember who continually talked during class and who did not when I decide whether or not to bump that grade up to an 'A'.***

-Put your Cell Phones on Vibrate – If you have an ***emergency*** call you should as quietly as possible leave the room and take the call. You should not take calls during class unless they are emergencies as this distracts the class.

-My thoughts on cheating – If you are cheating in class you **are wasting my time**, your time, and money (both your own money and that of the taxpayers). I also strongly believe that someone who would cheat in class

(regardless of whether they are the helper or helpee) would cheat in life and at work and is not someone I would recommend for a scholarship, job or for anything really. If you cheat on homework I WILL know even though I may not say so – what will happen is if you ask me for a recommendation or referral I will say no (just don't bother asking me for recommendations if you cheat on homework). Sometimes I suspect cheating on the exams and can't prove it, if that happens I will also say no to any recommendation or referral requests. I will do my best to prevent cheating at the front end by giving multiple versions of tests and locking computers and/or the internet and requiring phones are off - however as well all know a determined cheater will do anything they can to ruin it for everyone, in the end it's a waste for them and I feel sorry for their future employers.

X. Academic Honesty Policy

This instructor follows the Glendale Community College Honesty Policy as listed in the *Glendale Community College Catalog* and the *Student Handbook* (free at Information Desk near Admissions). Students are, at all times, required to do their own work. No copying of other students' work, whether on a test or on routine class work, is allowed at any time. Activities that are considered to be CHEATING include, but are not limited to, the following: communication with another person during an exam, accessing materials electronic or otherwise without the instructors express permission. Violation of any of these rules (i.e. cheating) could result in a lowering of the exam grade or the course grade (e.g. a "Fail"), and the violator's name and student I.D. number will be sent, with a description of the violation, to the Division Chair and to the Vice President of Instruction to be kept on record for future reference. The Dean of Student Activities may also be contacted for disciplinary action, if necessary.

Specific Activities that constitute cheating in this class

In this class I encourage you to discuss programming assignments and help each other during labs. HOWEVER, the following is not helping but cheating:

*TAKING PHOTOS OF EXAMS (really, what grade are we in?)
*Trying to "break" the network to cheat by stealing someone else's exam (it doesn't work and I can see you doing it, I will take a screen shot and send it to the Dean – she may suspend you from the college)
*Not sitting in your assigned seat during an exam.
*Copying directly off the person's computer next to you during an exam even though they have a different version of the exam and then telling me it is "coincidence" that your exam looks like the other version!!!! (there is no such thing as that type of coincidence)
*Having a phone or electronic device visible on your desk during an exam – if it's there I assume you're using it and cheating – NO EXCEPTIONS.
*Copying someone's program exactly or almost exactly (including homework – copy/paste is cheating not helping)
*Copying someone's program with only the variable names changed
*Using an assignment you got from a student that took this class in a previous semester as a guide for your own assignment or during a programming exam (even though this semester's assignments/exams are different this gives you an unfair advantage)
*Using a test from a prior year that another student stole.
ANY TALKING OR COMMUNICATING DURING AN EXAM FOR ANY REASON** – Doing this degrades the entire computer science profession and makes an 'A' mean Average (if "everyone" has an 'A' it really isn't that special any more). The 'A' student giving away the answers is the biggest problem in this regard. ***If you are warned to stop talking during exams do not contact me for references or recommendations after the class ends because I will NOT conscionably be able to give you one. In addition, if you cheat on all your homework or give your homework away to other students, I WILL KNOW, and if you ask me for a letter of recommendation or reference will not be able to conscionably give you one. In addition I will remember who talked/got warnings during exams when it comes to the end of the class and someone has an 89% I'm trying to decide to bump up to an 'A' or not.

NOTE: No two programs should ever look so much the same that as I'm grading I recognize pieces of code (maybe in the early assignments but definitely not in later assignments). The first time you will get a warning your assignment looks too similar to someone else's. The second time you will both get a zero on the assignment be reported to the business division chair and further action will be taken as she sees fit.

I am very flexible with programming assignments (they are only graded on REASONABLE EFFORT!!!) so if you are having trouble let me know and I will help. If I find many people in the class are having trouble I usually give extra time. In other words, you shouldn't have any reason to cheat.

I will be passing out a list of rules for both types of exams we have (objective and programming) sometime before the exams. This is a response to some problems I've had in past classes, so don't be nervous, I want everyone to learn and enjoying learning in this class; the rules won't be an issue as long as everyone does their own work**.**

******IMPORTANT TEST POLICY:** Seating will be assigned for the exams. During the class before the exam I will tell you which computer you will be required to sit at – the test will not start until everyone is sitting at their assigned computer (seat). If you fail to sit at your assigned seat or refuse you will not be allowed to take the exam.

XI. Academic Integrity

In this class I encourage you to discuss programming assignments in order to better understand the general concepts. Explicit copying of programs or assignments is not allowed but you may explain to someone and help them fix code (there is a difference between copy/paste and help on homework). NO HELPING WHATSOEVER IS ALLOWED ON TESTS – THIS SHOULD BE OBVIOUS. Make sure and include comments in your programs and show your work on your assignments so I know that you understood the work and that the work is your own.

XII. Students with Disabilities

All students with disabilities requiring accommodations are responsible for making arrangements in a timely manner through the Center for Students with Disabilities. If you have any questions feel free to email me and I will help you make the proper arrangements.

XIII. Issues or Concerns

Please address any issues you may have that are relative to this course *with me, your instructor*, either in person during my office hours (see above) or by email as early in the semester as possible (kshroyer@glendale.edu). I will do my best to resolve your concern. If you and I cannot resolve the issue, I will refer you to the division chair, Rory Schlueter, rory@glendale.edu , 818 240-1000, Ext. 5886 or see Seda Melikyan in the Division Office, SR 311, Ext. 5484, for an appointment.

XIV. Student Email Accounts

Please activate your GCC student email account; you can do this by going to MyGCC. This is one way the school communicates important information and if you don't activate your account you won't receive the information. This is also the primary way I will communicate with you via email. You can set up the account to forward to a different email you use for school to make it more convenient. ***In addition MOODLE sends all messages from me to your student email – you could miss important announcements (sometimes I give hints on what to study for exams) if you don't activate your account.***

XV. SLOs (Student Learning Outcomes)

This is just something administration requires we put on the syllabus in this specific format (much more complicated than the way it's already been explained above) – all of it is explained already in the earlier sections so this is just a repeat for you to meet requirements.

- Student will analyze a programming task from the perspective of a production programmer, with attention to clarity of code and comments and the interactions between the task and the computer environment.
- Student will critically evaluate the workability of a programming solution based on a knowledge of computing machinery internal operations.
- Student will recognize programming problems on both a function-by-function and an object oriented basis and develop structured code or objected oriented code based on the approach taken.
- Student will develop a firm foundation in programming, in addition to program efficiency, through the use of data design as a preliminary to program design and coding.