**Comp-Sci 201L: Problem Solving and Programming II - Lab**

**Fall 2016**

**LAB INSTRUCTOR:** Joshua Ford ([jdfxbb@mail.umkc.edu](mailto:jdfxbb@mail.umkc.edu))

**TIMES & LOCATIONS:** Tuesday 7:00 – 9:30

Flarsheim Hall 457

**DESCRIPTION:** Programming exercises and demonstrations to reinforce concepts learned in Comp-Sci 201 and provide additional practice in C++ programming. Prerequisites: Comp-Sci 101, Comp-Sci 191. Concurrent enrollment in Comp-Sci 201R required.

**MATERIALS:** *Absolute C++*, 5th edition by Walter Savitch

**ATTENDANCE:**

Attendance of the labs is mandatory. You are not allowed to change lab sections. Labs may not be made up unless an absence is excused. Excused absences include documented proof of illness, family emergency, work-related travel, and university-sanctioned travel (e.g. student organizations, athletic events, etc.) If you are uncertain if an absence will be excused or not, please discuss with your lab instructor ahead of time. You must let the lab instructor know about university-sanctioned absences at least a week ahead of time.

In the event that you are excused, it is ***your*** responsibility to contact the lab instructor about the missed assignment. Depending on the assignment, make up labs may be given to you to complete during another lab period. ***No makeup labs will be allowed after the week of the final lab.***

**LAB ASSIGNMENTS:**

The labs are designed to provide additional practice of new concepts with the guidance of the instructor before doing the homework assignments on your own. It is encouraged that you ask the instructor if you have any questions. Some discussion among classmates is fine; however, labs are to be worked on and turned in individually.

Lab overviews will be posted on Blackboard and emailed a few days before lab. The overviews will list the topics the labs will cover. At the beginning of class there will be a brief review over the material covered in lectures, and any questions can be answered. The assignment will be handed out. Typically most labs will be programs to be developed in Visual Studio. All work is expected to be uploaded by the *end* of the lab period.

**GRADING POLICY:**

All labs are worth 60 points each. All assignments will be posted to Blackboard and are to be turned in on Blackboard too. Grades will be updated on the My Grades section of Blackboard as the labs are graded. If grades are not posted, they have not been graded yet.

Assignments will be partially graded by computer. This means that your program’s output should match the expected output exactly (including spelling, grammar, punctuation, spacing, etc.). While sample input and output will be provided in class, different input and output data will be used when running the program, ensuring your code works as expected. If you have questions about the type of data being tested, please ask the instructor. Your code will be reviewed by the instructor for the remainder of the criteria below.

*Visual Studio assignments:* You will be given a problem to complete in class. Some template code may or may not be provided for you. You will be provided sample input and output code. Each programming assignment will be judged on the following criteria:

**Algorithm (10 points)** – The algorithm is the step by step logic behind your program. If implemented correctly, your algorithm should work for all test cases based into it. Some questions you should be able to answer are:

* Does your program perform all the steps required to successfully solve the problem?
* Does your program account for all possible input conditions, including upper and lower bounds?
* Does your program satisfy all of the criteria asked for in the assignment?

**Functional/structural composition (10 points)** – The functions, classes, or other structures in your code should be set up and named properly. Some questions you should be able to answer are:

* Are your functions, control structures, and variables set up correctly?
* Is code that is repeated broken up into functions instead?
* Are the names of your structures appropriate to what they do?

**Proper input/output (10 points)** – Lab programs will likely take in some form of input, and after doing some work, produce some form of output. Generally these labs will provide input via a text file and your will produce output also by text file. Some questions you should be able to answer are:

* Does the program read in the input correctly?
* Does the program’s output match the expected output exactly?
* Is there any extra data printed to the screen or file that is not expected?
* Is the user expected to do anything out of the ordinary?
* Do all input conditions produce the correct outputs?

**Documentation (10 points)** – Commenting your code is important. While you might know what you’re doing, anyone who reads your code may not as easily understand. In addition, you may forget in the future what you are doing. Comments for every line are not required, but someone should be able to tell what all parts of your program are doing from reading your documentation. Some questions you should be able to answer are:

* Is your name, date, and lab at the top of every file you turn in?
* Is the purpose of every class and function described appropriately?
* Are all major steps of your program described?

**Programming style (10 points)** – When writing your code, you should use a standard style throughout your code. For example, all variables should start with a lowercase letter, and for each word capitalize the first letter (ex: int numberOfCharacters = 0). You should also ensure all of your code is aligned on an even level, with code inside of blocks indented more than surrounding code. Some questions you should be able to answer are:

* Do all of your variable and function names match in style (lowercase first word, uppercase other words)
* Do all of your class names match in style? (Start with uppercase letters)
* Are all of your blocks (if statements, loops, functions, classes) indented properly to easily tell what code is applicable to which block?

**Compilation (10 points)** – Compiling or building your code often ensures that as you are writing code, it is error-free. Compiling often also saves your code and ensures if something happened you can recover your work.

* If your program compiles at all, you automatically get 10 points.
* If your program does not compile or gives any errors, the 10 points are lost.

Points will be deducted from each category for items that don’t match the grading criteria. Multiple issues will have multiple points deducted.

*Class participation:* You will be expected to follow along in class, participate in the discussions, and upload what we worked on together. Participating regularly will help you at the end of the semester if you are close to the next grade.

Grades are posted on Blackboard as soon as labs are graded. Check regularly to find out what your scores are. If you have an issue with how something is graded, please see the instructor about it. You have two weeks from the time it was graded to discuss a possible grade change. After that you may discuss it, but the grade will stand.

**EXAMS:**

There will be two lab exams, one mid-semester and one at the end of the semester. The goal of the lab exams are to test your knowledge of implementing a full program from start to finish on your own. Exams are worth double the amount of a normal lab. Exams will be an entire programming assignment to be completed in class. For reference, you MAY use your class or lab notes, previous homework assignments or labs, and your book. You may NOT use the internet, classmates, electronic devices (cell phones, tablets, laptops, etc.), or other sources. You must use the lab computer to do the exam, so copy any necessary files to a flash drive or the school network before starting. You may not leave the room during the exam period without permission from the instructor. There will be no final exam for this lab.

|  |  |  |  |
| --- | --- | --- | --- |
| Labs (12 assignment, 60pts ea.) | 720pts | | 75% |
| Exam 1 | 120pts | 240pts | 25% |
| Exam 2 | 120pts |
| Total | 960pts | | 100% |

**FINAL GRADES:**

Final grades will be assigned according to the following scale:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | 93% and above |  | B+ | 87% - 89% |  | B- | 80% - 82% |  | C | 70% - 74% |  | D | 60% - 64% |
| A- | 90% - 93% |  | B | 83% - 86% |  | C+ | 75% - 79% |  | D+ | 65% - 69% |  | F | 59% and below |

**UFIRST:**

UMKC Faculty and Staff are committed to assisting you as you work to achieve academic success. UFirst is an early alert program that allows your course instructor or other university staff to issue early alerts if they become concerned about your successful academic progress or if you express a concern to them regarding your ability to achieve success at UMKC. As a follow up to the early alert you may be contacted by an academic adviser or other university support staff to provide you with information, resources, or referrals to help you address the concern, and to provide you with additional support to assist you in achieving the highest possible level of academic success while working to complete your UMKC degree. For more information, go to <http://ufirst.umkc.edu/>.

**POLICY ON ACCESS FOR STUDENTS WITH DISABILITIES:**

To obtain disability related accommodations and/or auxiliary aids, students with disabilities must contact the Office of Services for Students with Disabilities (OSSD) as soon as possible. To contact OSSD call 816-235-5696. Once verified, OSSD will notify the course instructor and outline the accommodation and/or auxiliary aids to be provided. For more information go to: <http://www.umkc.edu/disability/>.

**POLICY ON STUDENT CIVILITY:**

The department’s commitment is to create a climate for learning characterized by respect for each other and for the contributions each person makes to class. We ask that you make a similar commitment.

Please turn off and store all personal electronic communication devices, including cell phones, during the labs.

Since this class takes place in a computer lab, we expect that you will NOT use the computer to conduct any non-class-related activity during class time. The computers are to be used for in-class lab work only. If a student is found to be surfing the web or engaging in any non-class activity on the computer during class, a zero will be assigned for the lab.

**POLICY ON STUDENT CONDUCT:**

Adherence to the Student Conduct Code is expected. A student enrolling in any UMKC course is expected to exhibit high standards of academic honesty. In the case of academic misconduct, the instructor will assess the affected work and report the incident according to the guidelines printed in the catalog. The Vice-Provost may impose additional sanctions ranging from "Warning" to "University Suspension" to "University Expulsion." The rules governing any suspected violation can be located at <http://www.umsystem.edu/ums/rules/collected_rules/programs/ch200/200.010_standard_of_conduct>.

The work you do for all projects is expected to be your own effort. You may share some ideas with others but your program or implementation must be your own. For example, it's all right to debate with others the advantage of different implementations, but you must not view or copy code from anyone else.

**You may NOT reference code from another student in this class or in any previous version of this class. You also may NOT reference code from any source online.**

If this should occur, all students involved in the plagiarism will receive a zero for that assignment. If you have any question about your level of collaboration, you should ask.