## Course Syllabus

## Course Title: Programming Logic \& Technique, CIS1400-003 FL 2023

Credit hours: 4 -Clinical Hours: 0 Lecture Hours: 4 Lab Hours: 0 VC
Instructor: Robert Burrows - for a bit about me see https://www.cod.edu/faculty/websites/burrows/index.aspx E-Mail: burrows@cod.edu Every attempt will be made to answer e-mail on a daily basis. When sending emails please clearly indicate the problem or concern you are having, your name and course enrolled. If you have a programming problem you can't figure out (after spending at least 30 minutes trying $;)$ ), I will have an alert set up so that if you enter "1400-HELP" in the subject line of your email I will get paged and get back to you as soon as possible. All communications between us will be through the COD email and/or blackboard possibly in an online help session.

Office Hours/additional help: By request in class or via email- there will be periodic help sessions arranged outside of class as requested by students

Dates, Day \& Time, Classroom: 08/24/2023-12/14/2023 Lecture/Discussion Thursday 06:00PM - 09:50PM, Seaton Computing Ctr, Room 115

## Textbook \& Materials Needed:

Starting Out with Programming Logic and Design, $6^{\text {th }}$ Ed. By Tony Gaddis, 2019 ISBN: 978-0-13-760214-8, Publisher: Pearson.

You can get a copy of the book at the COD bookstore, Follet Discovery link, https://www.bkstr.com/dupagestore/home/en The online version of the book is $\$ 39.96$ which I recommend. If you like a printed (hardcopy) of the book to keep as a reference and make notes in as you read, it costs $\$ 149.98$ new and $\$ 112.50$ used. Make sure you get the $6^{\text {th }}$ edition as some earlier versions maybe free but will not work for this class.

You will need a PC or MAC for the class for homework. If you don't have a computer and cannot afford one at this time the college library can lend you one. See request form on the Library website.

We will use the Python programming language in this class. Python is a relatively easy language compared to Java and C++. Python is a free download and the current version used for class is 3.11 .4 which you can download to your Mac or PC at https://www.python.org/downloads/ The install will detect if you have a PC or MAC. If you need help with the install please email me and I will help you.

## Your and My Goals for the class:

I will ask for your goals and get to know you in the first class assignment.
My goal is to help you reach your goals in any way I can; and since retired from my day job I have lots of time to help.

Especially in these strange times of pandemics, climate change, and racial injustice to name a few, statistics have shown a growing number of students experiencing mental health challenges to varying degrees. Doing what you can to stay ahead and on top of depression or anxiety by wisely taking care of yourself will be a key
to succeeding academically. But even then, sometimes these challenges can affect your ability to complete the required work. Or a particular assignment might trigger anxiety for you in ways I have not anticipated. Or maybe you reach a point where you just can't get yourself to class at all.

In any of these cases, please come and talk with me or at least send me an e-mail. I'll listen and do what I can to help. The sooner you share your challenges with me, the more I can help you plan to succeed in this course. To learn the material and pass the course or earn an A, you'll still need to do every bit as much work as other students, but we may be able to find some creative ways to help you do that-especially if you approach me when your problems arise, instead of at the end of the semester.

## Course Description

An introduction to computer-based problem-solving techniques. Includes software design tools such as structure charts, Input Processing Output (IPO) charts, flowcharts, pseudocode, and Unified Modeling Language (UML) diagrams. Concepts such as documentation, structured design, modularity, Object Oriented Program (OOP) design, and event driven programming are covered. Programming of algorithms are implemented using a high level language that emphasize structured and object oriented design techniques. Repeatable for credit: No Pre-Enrollment Criteria:
Prerequisite: MATH 0482 Foundations for College Mathematics II with a grade of "C" or better, or equivalent or
Prerequisite: MATH 1115 Technical Mathematics I with a grade of "C" or better, or equivalent or a qualifying score on the mathematics placement test or Consent of Instructor

Topical Outline: I have divided the class into 6 learning modules - divide and conquer
Module 1 - Introduction + chapters 1 and 2 in your book

1. Program Development Lifecycle (PDLC)
2. Software design techniques
3. Computer based paradigms
4. Simple data types
5. Variables, constants, and literals

Module 2 chapters 4 and 5 in book (Note we skip ch 3 and as a result skip some reading in ch $4 \& 5$ )
6. Control structures
a. Sequential
b. Selection
c. Repetition
7. Arithmetic, relational, and logical operators

Module 3 chapter 3 and 6 in book.
8. Local and global variable scope
9. Modularity, parameter passing, and return values
10. Input validation

## Module 4

11. Advanced data types: Arrays
a. Single dimensional
b. Multi-dimensional
c. Parallel
12. Searching and sorting algorithms for arrays

## Module 5

13. Files
14. Menu Driven Programming
15. Text Processing

## Module 6

16. OOP design concepts
17. GUI and event driven programming

## General Course Objectives:

Upon successful completion of the course the student should be able to do the following:

1. Explain steps used in program development cycle
2. Identify tools used in software design
3. Create algorithms to solve both verbal and written problems
4. Differentiate simple data types
5. Differentiate variables, constants, and literals
6. Apply concepts of structured program design such as modularity, sequence, selection, and repetition
7. Differentiate arithmetic, relational, and logical operators in algorithm design
8. Demonstrate variable scoping in program design for local and global variables
9. Apply data transfer techniques between modules using parameters and return values
10. Construct applications to use files for input and output
11. Implement arrays as structures to contain data
12. Use searching and sorting algorithms in problem designs
13. Use a higher level programming language to code, test, and debug software designs
14. Implement concepts of abstraction and encapsulation using Object Oriented Programming (OOP) design
15. Explain advanced OOP design techniques such as inheritance and polymorphism
16. Describe integration of Graphical User Interfaces (GUIs) and event driven programming

## Course Requirements

Student Responsibilities: Keeping up with class assignments is critical in successfully completing this course. If you have conflicts with personal or work issues please let me know as soon as possible.

In addition, during the pandemic, being in the classroom you must comply with current school Covid requirements found at https://cod.edu/coronavirus/index.aspx or you will not be allowed in the classroom; and if not compliant within two weeks you may be dropped from the class. Finally, if you get Covid, please let me know ASAP and we will make accommodations.

Class attendance and active participation are essential if students are to receive maximum benefit from the class. Participation requires preparation including completion of reading, labs, assignments, assessments, and exams by the due dates. All due dates will be mentioned on the class blackboard. Students are responsible to submit all work on or before the due dates. It is the students' benefit to use their time wisely whether it is in
preparation for class, during scheduled class, or in the lab. When students are in any COD lab environment, they should abide by the college policies. Questions, comments, and suggestions are welcome.

Exams and make-up policy: There will be two exams. No retakes of exams are allowed. Make-up exams are not encouraged except on emergency situation.

Assignments (including weekly quizzes): It is extremely critical that students complete all assignments on time. Past experience has shown that students that were behind on their assignments never caught up. Students may submit assignments early. Submitting assignments in the order assigned will ensure progression according to academic design of the course. If an assignment is late you will start with half credit.

Help in person on campus: I will be able to meet with you on campus or online for one on one or class help sessions. Please send me an email or arrange with me in class. Before and after class will save us both a trip to the college -

Academic policy: Any violation of COD policies regarding academic honesty and/or integrity will be referred automatically to the appropriate college authorities for disposition. Please see appropriate pages in the college catalog for definitions and regulations. The minimum penalty for cheating will be a zero for all parties involved on that exam, assignment, lab, or assessment. AI (chatGPT) is not perfect, but can be a tool to help.

Withdrawal policy: Missing three or more assignments or quizzes before mid-term without a reason discussed with me will result in the student withdrawn from the class for non-Pursuit of the course objectives. No longer attending a class does not constitute an automatic withdrawal. All withdrawals must be done on or before Sunday November 12th as in the College's academic calendar (https://catalog.cod.edu/academic-calendar/ ) It is important to drop if you cannot complete the work as an Incomplete grade is not allowed in this class.

Class Behavior and set up: Class participation is important, during class time, considerate conduct by all persons is important to a favorable learning environment. Any infringement on the rights of others to get education will be dealt with in an appropriate manner. Please to avoid distractions during class and have your computer on mute unless asking a question.

Team Projects in class. Being a good team player is an essential for a programmer on the job. You can pick or I will assign you teams to work with on projects in class. If due to Covid concerns or other issues you do not want to work directly with your classmates, let me know and you will be able to work in an online team.

General notes: In order to achieve the course objectives, it is essential that comply with the above requirements, the rules and policies of COD contained in the catalog and other COD materials.

Class period rough outline - may change slightly week to week
5:30 - 6:00 Help session as needed Note if you would like to come in earlier email me
6:00-6:05 Questions from last class
6:05-7:50 Lecture/discussion - can take break whenever need to
7:50-8:00 Break
8:00-8:50 Lecture/discussion
8:50 - 9:50 Homework help
10:00 - 10:30 After class Homework/question session as needed

## Grading and Evaluation Criteria

Your final grade is based on accumulation points of all class work including homework, quizzes, and exams.
NOTE: Due dates for all homework and Quizzes are as assigned in blackboard. Points are as follows:
\(\left.$$
\begin{array}{|l|l|}\hline \text { Programs/Assignments } 25 \times 10 \text { pts/each and } 1 \times 20 \text { pts (family recipe program) } & 260 \\
\hline \text { Quizzes } 13 \times 10 \text { points each - checks to insure you understand the material in the book } & 130 \\
\hline \text { Tests (50 points for midterm and } 70 \text { points for Final) } & 120 \\
\hline \text { Total points on assignments and tests } & 510 \\
\hline \begin{array}{l}\text { Extra Credit - there will be additional programs/assignments for extra credit throughout the } \\
\text { semester. You can also earn extra credit points for class participation at my discretion. Extra credit } \\
\text { (EC) assignments fall into } 3 \text { categories and may not be available to all students as described below: }\end{array} & \begin{array}{l}\text { Up to } \\
\text { 1) } \\
\text { EC available for all students and will be denoted as such. }\end{array}
$$ <br>
2) EC are only for those students who are struggling with the material or behind in their work which <br>

include the checkpoint extra credits which must be in by their due date.\end{array}\right]\)| 3) EC are only for those students who have kept up with the work and want to be challenged |
| :--- |
| beyond the course material presented. Some of these students may have already had a |
| programming course. |

Final Grades will be assigned using the following point scale which is on a $90 \%, 80 \%, 70 \%, 60 \%$ and below out of the 510 points before extra credit.
Notes on grades:

1) Blackboard shows extra credit in our pct grade, in other words if extra credit is 5 points and you haven't done it black board gives you 0 out of 5 points which makes your pct grade in blackboard appear lower but just ignore this percent grade in BB. Your grade is just based on points and if you don't do extra credit you don't lose any points - extra credit can only help you.
2) Your grade is simply based on your point total. Please do not ask for a higher grade at the end of the semester even if you miss the next grade by 1 point.
3) After the final exam you will see your point total and have a day or so to complete assignments and/or do extra credit to move your grade up.

| Accumulated Points | Grade | Percentage |
| :---: | :---: | :---: |
| $459-550$ | A | $>=90 \%$ |
| $408-458$ | B | $80-89$ |

8/15/23

| $357-407$ | C | $70-79$ |
| :---: | :---: | :---: |
| $306-356$ | D | $60-69$ |
| 305 or below | F | $<60 \%$ |

## Course Outline

1) The outline below is tentative and may change during semester depending on how our class progresses 2) Assignments, Assessments, and hands-on labs will be available on the class blackboard (BB) site. The due dates for assignments below are also in the BB course calendar which should show all your classes' assignments - Drop down by your name on upper right of blackboard. All Python submissions should include your Python program(s) as a .py file and test case output in a .txt file
2) The dividing line between each class shows what we will cover before break and after break. 4) All assignments are from the end of each chapter in your textbook in the Programming Exercises section. For example, Prog EX 2-6 Sales Tax means Programming Exercises Chapter 2 exercise 6 on page 99, and NOT exercises 2 through 6, nor the example 2-6 on page 44 . Note the online textbook does not have page numbers so to find the assignment but in the name, e.g. Sales Tax for exercise 6 in chapter 2.
3) All assignments not turned in by due date in last column of the table below will start with half credit.

NOTE: If you email me with valid reason why you are late or why you are going to be late and when you plan to complete the homework or quiz, at my discretion you may still get full credit. As a programmer or software developer in the real world, it never hurts to ask for additional time to finish a program
6) Reading is not submitted as an assignment. Your reading is checked by the chapter quizzes which has multiple choice questions on the reading.
7) Quizzes not submitted by the due date you will start with half credit. Again, my discretion to extend due date for you if you email me with a real issue and a plan date to do it.
7) Active reading.. Your textbook solves problems with pseudocode. As we will go over in class, pseudocode is description of your program logic without the formal syntax of a specific programming language. From pseudocode you can then write your program in any language. As mentioned, the specific language we are learning in this class is Python. So, I am going to give you a Python Supplement guide for each chapter in your textbook (starting with chapter 2) You can find these supplement guides in BB under Python Supplement link. As you read the pseudocode examples in each chapter you should also read the Python companion and see and try the same pseudocode examples in Python. An example follows:

When reading Program 2-10 on page 53 (online readers can search for 2-10) shown below:

```
Program 2-10
1 Display "Enter the first test score."
2 Input test1
3 Display "Enter the second test score."
4 Input test2
5 \text { Display "Enter the third test score."}
6 ~ I n p u t ~ t e s t 3
7 \text { Set average = (test1 + test2 + test3) / 3}
8 \text { Display "The average score is ", average}
```

If while reading you are following along in the Python supplement (found in BB) for chapter 2:

You will see the Python program corresponding to this book pseudocode above as:
In Python this program is (assuming test scores are integer numbers):
test1 = int(input("Enter the first test score."))
test2 $=\operatorname{int}($ input("Enter the second test score."))
test $3=\operatorname{int}($ input("Enter the third test score."))
average $=($ test $1+$ test $2+$ test 3$) / 3$
print("The average score is ", average)
\# comment- this output also be written as print(f"The average score is \{average\}") Try it
So to be an active reader you can:

1) Got to IDLE and do a File - New File to create a new program.
2) Copy and paste the program (starting with test $1=$ ) from the supplement shown above and paste into IDLE
3) Run the program in IDLE Run- Run Module
4) IDLE will ask you to name program and save it - name it something like Program 2-10 so you can refer back to it iif needed
5) Enter some test data (I'll talk about simple input - like below that the average is 90)

Enter the first test score. 80
Enter the second test score. 90
Enter the third test score. 100
The average score is 90.0
6) Next try changing the print statement to the "f string" form that I mentioned in class was more popular $\operatorname{print}(f$ "The average score is $\{$ average $\}$ ")
and as it says in the supplement "Try it" you may use the same input or different input
You also may try things on your own (experiment! ) like change the period(.) in each input prompt to a colon (:) and run the program again

This is what I mean by active reading with the supplement
The Python Supplements also covers the syntax of the Python language and review some of the Python language I present in lecture. . Please let me know any errors you find in the companion guide you will get extra credit for errors you find
Reading assignment of the textbook and the companion guide chapters are due for the class as in the right column below.
Again, You should be able to see these assignments in your Course Calendar as described above, along with the assignments of other classes if you are taking any.

| Module <br> Class\# <br> Date | Topics covered in class <br> And homework/quiz start dates | Due Dates <br> Reading textbook homework, quizzes. And extra credit |
| :---: | :---: | :---: |
| Mod 1 <br> Class 1 <br> Aug 24 | Course introduction <br> Programming and algorithms <br> OTTO - types of programming logic <br> Review syllabus <br> Install Python <br> Class student Survey to determine your experience objectives, and motivation <br> HW 1: Student Profile in class <br> Textbook Reading <br> Chapter 1 Introduction- a look inside the computer <br> HW 2: Prog Ex 3-1 | Read Ch 1 book <br> Note: No Python Supplement for Ch 1 |
| $\begin{gathered} \text { Class } 2 \\ 21 \end{gathered}$ | Chapter 2 Input, Process, Output - variables <br> Pseudocode \& Flowcharts <br> Example: Prog EX 2-8 IPO, flowchart, pseudocode, and Python in class <br> Chapter 2 more examples, review and questions. <br> HW 3: Prog Ex 2-6 <br> Quiz Chapter 2 | Actively Read Ch 2 book and Python companion guide. For rest of syllabus read chapter means both the textbook and companion guide. <br> HW 1 <br> HW 2 <br> Extra Credit - want to be a programmer? <br> Extra Credit Saundra McGuire's Video on being the best student you can be |
| Class 3 <br> Sept 7 | SPECIAL GUEST <br> Chapter 2 finish up <br> Do Lab 1-6 Pedometer.docx found in Week 3 <br> notes and materials <br> Prog Ex 2-15 Ingredient Adjuster 101 book <br> Structure and Boolean Logic <br> Lab 1.6 Pedometer Calculator pseudocode <br> Chapter 3 If statements - Decision <br> HW 4: - recipe program. Pick your favorite or family favorite recipe and write a program very much like Ex 2-15 Ingredient Adjuster. You should turn in 3 attachments as shown in the assignment | Read Ch 3 <br> Quiz Ch 1 <br> Quiz Ch 2 <br> Extra Credit Lab 1.6 Pedometer Calculator turn in Python code |
| Mod 2 | Chapter If statements - Decision Example: <br> Program Ex 3-2 \& 3-5 pseudocode and flowchart in class <br> HW 5: Prog EX 3-1 <br> Quiz Chapter 3 |  |
| $\text { Class } 4$ <br> Sept 14 | Review Point Total <br> Finish Chapter 3 <br> Another decision logic flowchart example Chapter 4 Repetition Structure (looping) finding a parking spot | Mod 1 CheckPoint Extra Credit for those behind Extra Credit Python for batting average program HW 3 <br> Read Ch 4 |


| QTR grades | HW 6: Prog Ex 3-6 <br> Chapter 4 continue on looping <br> Prog Ex 4-1,4-2 in class Flowchart and <br> Python - 4-5 flowchart in class - python extra credit and/or 4-9 Pennies for Pay <br> Pub Crawl flowchart. <br> --Review Your Point Total-- <br> Quiz chapter 4 <br> HW 7: Prog EX 4-2 | HW 4 <br> Quiz Ch 4 <br> HW 5 |
| :---: | :---: | :---: |
| $\begin{gathered} \hline \text { Class } 5 \\ \text { Sept } 21 \end{gathered}$ | Finish up chapter 4 HW 8: Prog EX 4-3 | Read Ch 5 <br> HW 6 <br> Quiz Ch 4 <br> HW 7 |
| Mod 3 | Chapter 5 Modules - Introduction Divide and Conquer Make programming exercise 2-8 modular Python Quiz chapter 5 <br> HW 9: Prog EX 5-2 Sales Tax modular |  |
| Class 6 <br> Sept 28 | Finish up Chapter 5 <br> Chapter 6 Functions <br> Quiz Chapter 6 <br> HW 10: - Prog EX 6-1 <br> Chapter 6 Functions passing variables in class design game of Craps - a true need for a flowchart to see logic Built in functions Turtle package <br> HW 11: Prog EX, 6-7 | Mod 2 CheckPoint Extra Credit for those behind Extra Credit Program 4-5 tuition in Python <br> Extra Credit Program 4-9 pennies Python <br> Read Ch 6 <br> HW 8 <br> Quiz Ch 5 <br> HW 9 |
| $\begin{gathered} \text { Class } 7 \\ \text { Oct } 5 \end{gathered}$ | Chapter 6 finish up and review <br> HW 12: Prog EX 6-12 <br> Chapter 7 Input validation - only 13 pages <br> Prog EX 7-5 in class - edit Rock, Paper, <br> Scissors <br> HW 13: Prog EX 7-1 <br> No Quiz for chapter 7 <br> Practice Midterm handout to try chap 1-7 <br> NOTE: no homework/quizzes accepted after midterm | Read Ch 7 - (only 10 pages) <br> HW 10 <br> Quiz Ch 6 <br> HW 11 <br> Extra Credit Python Turtle design <br> Extra Credit Python program for craps |
| Class 8 <br> Midterm <br> Oct 12 | Review practice midterm <br> Other Questions on chapters 1-7 <br> Midterm chapters 1-7-50 points 6:00-10:30 PM | HW 12 <br> HW 13 <br> Midterm Exam <br> Mod 3 CheckPoint Extra Credit for those behind <br> Extra Credit - get BB - Week 7 p341-342.py - 5 points extra credit if can make program not fail if "abc" entered. |


| $\begin{gathered} \text { Mod } 4 \\ \text { Class } 9 \\ \text { Oct } 19 \end{gathered}$ | Go over Midterm - 5 minutes <br> Arrays <br> Chapter 8 Array/Lists and Flowcharting. Goes with Repetition - loops and arrays like pizza and beer or pizza and pop. Prog EX 8-3 in class Group exercise <br> Help on Lottery program <br> --Review Your Point Total-- <br> From here and for the rest of class just submit <br> Python with test cases for homework <br> Quiz Chapter 8 <br> HW 14: Prog EX 8-2 <br> More Arrays Prog EX 8-3 in class parallel arrays <br> TIC-TAC-TOE <br> HW 15: Prog EX 8-6 | Read Ch 8 |
| :---: | :---: | :---: |
| $\begin{gathered} \text { Class } 10 \\ \text { Oct } 26 \end{gathered}$ | Finish up arrays <br> Chapter 9 Sorting and Searching Arrays <br> Swapping 2 array elements <br> Quiz Chapter 9 <br> Finish up Chapter 9 and review <br> HW 16: Prog EX 9-2 <br> HW 17: Prog EX 9-4 | Read Ch 9 <br> Quiz Ch 8 <br> HW 14 <br> Extra Credit PROG EX 8-3 Rainfall months with highest and lowest |
| $\begin{gathered} \text { MOD } 5 \\ \text { Class } 11 \\ \text { Nov } 2 \\ \text { NOTE } \\ \text { CLASS } \\ \text { ONLINE } \end{gathered}$ | Do Not come to COD - Online with collaborate ultra <br> Chapter 10 Files <br> Prog Ex 10-1 <br> Examples of Python programs with sequential <br> ASCII files <br> Quiz Chapter 10 <br> Finish up Chapter 10 and review <br> HW 18:Prog EX 10-1 <br> HW 19: Prog EX 10-2 <br> --Review point total -- | Read Ch 10 <br> HW 15 <br> Quiz Ch 9 <br> HW 16 <br> HW 17 <br> Mod 4 CheckPoint Extra Credit for those behind Extra Credit Tic Tac Toe ( I will provide part of program and go over in class) |
| $\begin{gathered} \text { Class } 12 \\ \text { Nov } 9 \end{gathered}$ | Chapter 11 Menu Driven programming and debugging Example 11-4 and <br> Book example on Metric Conversion <br> Student scores example <br> Quiz Chapter 11 <br> HW 20: Prog EX 11-1 <br> Chap 12 Text Processing with String data Example Prog EX 12-2 scan for periods "whitespace" Flowchart Finish Text Processing NOV 13th last day to withdraw <br> HW 21: Prog Ex 12-3 <br> HW 22: Prog EX 12-1 <br> Quiz Chapter 12 | Read Ch 11 <br> Read Ch 12 <br> Quiz Ch 10 <br> HW 18 |


| $\begin{gathered} \text { MOD } 6 \\ \text { Class } 13 \\ \text { Nov } 16 \end{gathered}$ | Chap 14 OOP <br> 14-2 algorithmic workbench <br> 14-1 Pets example get you started <br> Lynda Video on OOP <br> Quiz Chapter 14 <br> Chap 14 OOP and review Prog EX: 14-1 in class HW 23:Prog EX: 14-1 | Read Ch 14 <br> HW 19 <br> Quiz Ch 11 <br> HW 20 <br> Quiz Ch 12 <br> HW 21 <br> Mod 5 CheckPoint Extra Credit for those behind Extra Credit code example program 11-6 Coffee Inventory in Python |
| :---: | :---: | :---: |
| Nov 23 <br> No Class | Thanksgiving Holiday |  |
| $\begin{gathered} \text { Class } 14 \\ \text { Nov } 30 \end{gathered}$ | Chap 14 Finish <br> HW 24: Prog Ex 14-2 <br> Perhaps bit more on OOP <br> Chapter 15 GUIs <br> Review point total <br> GUI Example of Metric conversion program Practice Exam given out | Read Ch 15 Quiz Ch 14 HW 22 |
| $\begin{gathered} \text { Class } 15 \\ \text { Dec } 7 \end{gathered}$ | Chapter 15 GUI and review <br> Quiz Chapter 15 <br> HW 25: Prog EX 15-1 in class <br> HW 26: Prog EX: 15-3 <br> Practice Final Exam review for final Help to complete all homework but should have almost all done now | HW 23 <br> HW 24 <br> HW 25 in class <br> Mod 6 CheckPoint Extra Credit for those behind can schedule help session before final exam |
| Class 16 <br> Final <br> Exam <br> Dec 14 | Final Exam 6:00-10:30 as per COD schedule | HW 26 <br> Quiz Ch 15 <br> Extra Credit figure $15-8$ p 721 add formatting to dollar amount and input validation like in program 15-3 shown in class <br> Extra Credit prog Ex 11-1 Language translator as GUI - very similar to metric example shown in class |
| Dec 16 ${ }^{\text {th }}$ | Final Graded by 5:00 PM 12/15 so you can see if you may want to do extra credit for a higher grade Complete all assignments and extra credit - all work in - Dec 16 ${ }^{\text {th }} \mathbf{1 1 : 5 9 ~ P M ~}$ I will be available in school for help Friday or Sat if needed |  |

## Course Expectations

1. All assignments, (including assessments, projects, labs, and any related course work) must be submitted on or before the given due dates for full credit - see 2 . below
2. Late assignments are not acceptable unless on medical (with a doctor's letter) or emergency situation (with proof).
3. When a quiz or assignment due date has passed, it may disappear from the class Blackboard $(\mathrm{Bb})$. Please do not ask the instructor to reset the assignment due date or inform the instructor that you can no longer see the assignment on the Bb .
4. To receive full points, a complete assignment must be submitted by following the assignment's requirements and specifications.
5. Unless requested by the instructor, no assignments should be sent to the instructor's email address. Any assignments sent to the instructor's email without permission will be disregarded.
6. All assignments must be submitted on the class BB. For programming courses, all programs source code listings must be presented with the program output/result.
7. Submitting assignments in the order assigned will ensure progression according to the academic design of the course.
8. Discussions and collaborations are permitted. However, you must do your own assignments to learn and master the material. Dishonest work will be disciplined according to the university's policy.
9. If you are caught copying someone's work, you will be placed in one of the following possibilities (depending on the work):
a. You get a zero for your current assignment.
b. You will get one letter grade less in the course, i.e., if you get an A, you will get a B
c. You will be expelled from the university.

If you are caught copying above, all of your previous work relating to the course will be reexamined and re-evaluated. All of your future work in the course will be closely scrutinized.
10. You understand the grading criteria as mentioned in the syllabus above. The grading distribution will not be curved.
11. Your attitude and behavior relating to the course will be considered when a borderline grade befalls to boost up your grade via extra credit class participation to the next higher letter grade. Your grade is based on point scale, and extra credit after the final is the only way to get to the next higher grade.
12. An incomplete grade is allowed only in last two weeks of class due to an unforeseen event. Being behind on homework is not an unforeseen event.
13. You are responsible for withdrawing from the course if no longer attend the class. Please check the university's academic calendar on the last day allowed to withdraw from the class.
14. Bad planning on your part is not an emergency on the instructor's part.

