College of Southern Nevada
Course Syllabus
CIT137 – Special Topics: Programming languages
The official syllabus with the official schedule of activities will be available in Canvas on the first day of class.
This syllabus should give you a good idea of the CIT course. It has textbook information and a tentative schedule of activities.

A. Course Information	CIT137 - Special topics course that explores a programming language at an introductory level. Topics will include the language's data types, input, output, operators, decisions and looping statements, functions and other topics specific to the language. This course may be repeated by students who wish to explore up to three different languages, for a maximum of 9 credits. Prerequisite: IS 115 or Instructor approval. **Topic: Beginning Python Programming**
B. Course meeting time/days/ location	Online
C. Instructor Information	Name: Efatsadat (Eve) Taghva Phone: (702) 651-4403 Office Hours/Location: Office hours conducted via the Conference option in Canvas - TBA Google Voice number (talk & text) 505-652-2628 Email: Canvas Learning system – Alternate: efat.taghva@csn.edu Web site: https://bellagio.csn.edu/~etaghva/courses/ Office Mailbox code: CYA 2769
D. Learning Outcomes	By the end of this course, a student will be able to: 1. Solve problems using the fundamental syntax and semantics of the language. 2. Create programs that include appropriate loops, decision structures, data structures, and modularization. 3. Use generally accepted principles of good programming style and documentation. 4. Create programs that include files for input and output. 5. Create programs that include error handling.
E. Textbook	A textbook is not required
F. Late Work policy	I do not accept late assignments, nor do I provide makeup assignments. You are responsible for your own Internet connection when working remotely. Work ahead if you know you are going to have conflicts or time constraints. Please schedule yourself accordingly. There are NO MAKEUP

	exams or quizzes.
G. Method of Evaluation	Grades are based on exams and projects assigned throughout the semester. All assignments will be submitted via the Canvas assignment dropbox. Detailed instructions will be provided in the text of the requirements for each of the assignments**.** If the Canvas email system fails and you must use a different system to submit your assignments, send your assignments files to my alternate email address. All exams are delivered through the Canvas system. All exams will have strict time limits and detailed information will be provided at least a week before the date of the exam.
H. Grade determination	Your grade is based 2 exams, chapter quizzes, and homework assignments. Midterm Exam - 20% Final Exam - 20% Chapter quizzes - 10% Assignments - 50% All exams may include True/False, Multiple-choice, short answers, and essay type questions. Full details will be provided a week before the start of an exam. The following is how letter grades are assigned based on total percentages of assignments and exams.
I. Attendance Policy	College assumes maturity, seriousness of purpose, and self-discipline for meeting the responsibilities associated with each course. If you will need the instructor to sign documents testifying about your attendance, **YOU** must come to the instructor after each class you attend to let her know you were there. Class participation is a strong aspect of this course and your participation is always encouraged.
J. Academic Integrity	CSN demands a high level of academic behavior. Acts of academic dishonesty including plagiarism and cheating are regarded as very serious offenses. **Cheating will not be tolerated. DO NOT collaborate with anyone on individual assignments. If any duplicated work is submitted, all parties will receive 0 points for the assignment. On a second offense, the student will receive a grade of F for the course and may be subject to expulsion from school. You SHOULD NOT attempt to pay anyone to complete your code. This is a serious offense leading to immediate expulsion from the course and possibly from the college.**

100 - 94	A		**70 - < 77**	C
90 - < 94	A-		**60 - < 70**	D
87 - < 90	B+		**< 60**	F
84 - < 87	B			
80 - < 84	B-			
77 - < 80	C+			

	Scholastic dishonesty will not be tolerated. You are expected to have read and understood the CSN Academic Integrity Policy may be found at: http://archive.csn.edu/sites/default/files/u12821/academic-integrity-policy.pdf
K. Disability Resource Center	If you have a documented disability that may require assistance, you will need to contact the Disability Resource Center (DRC) for coordination of your academic accommodations. The DRC is located in Student Services on each major campus. More information about the CSN DRC please visit: https://www.csn.edu/drc For more CSN Americans with Disabilities Act (ADA) information please visit: https://www.csn.edu/ada
L. Disclaimer	This syllabus is subject to change with advance notice. Notices will be posted in the online forum. It is your responsibility to stay informed.
M. Tentative schedule	Please look at the end of this document for the tentative course schedule of activities.
N. Student Rights & Responsibilities	It is your responsibility to be aware of your rights and responsibilities. This information is located in the General Catalog and Student Handbook, which can be found on the CSN Catalog/Schedule/Calendar web page: https://www.csn.edu/sites/default/files/u2241/studentrightsrespsection.pdf
O. College Library Services	The Library offers a wealth of resources to help you with your research projects. There are libraries at each of the main campuses and an extensive collection of resources available from the Library's Homepage: library.csn.edu A note from your library: The library holds many workshops such as "College Library Services offers ongoing research workshops throughout the semester. Bring your topic or assignment to one of The workshops on the basics of locating and citing quality information and receive in depth assistance with a librarian. Check out the schedule at http://csn.libcal.com/calendar/events/ or call 651-5729 for more information."
P. References	**CSN Library Services** offers extensive in-person and online resources to help you complete assignments, including research and citation workshops, online articles and books, and drop-in research assistance at the Reference Desk inside each campus library and online at library.csn.edu "Safari Books Online" is of special interest to students in CIT courses. To access this digital library of technical books and videos, click on the Browse Databases button on the Library's homepage, then click on the letter S to filter the databases. The link to "Safari Books Online" should be at or near the top of the resulting list. Initially, you will have to enter your student email address, then create an account with Safari. Subsequently, you will use your email address and your Safari password to access the Safari resources.
Q. Required extra- or co-curricular activities	All activities are based on projects and exams assigned throughout the course. Any required extra activities will be clearly explained in class.

R. Safety	This class does not have an experiment lab and therefore we will not be concerned about following specific safety strategies.
S. Additional fees	There are no additional fees for this course.
T. Additional Information	
CSN Student email	All students enrolled at CSN have a CSN Student Email account. Beginning February 1, 2020, all information from the college will be sent to your CSN-issued student email address (enrollment information, financial aid information, cashier information, college events, etc.). **It is extremely important that you check your student email daily.** You can access your student email through GoCSN (go.csn.edu). Once you validate your student email address you will have access to Microsoft Office 365 for up to five devices and 1TB of OneDrive storage. www.csn.edu/email
Important Note	If you have any concerns about this course and/or me, please contact me first. If I cannot resolve your issue, please contact Arlene Menezes in the CIT Department Office at 702-651-5976. You will be directed to the appropriate Program Director or the Department Chair. You will remain anonymous, if possible, and all communications will be strictly confidential. Please **DO NOT** wait until the last minute to make your concerns known to me and/or to the CIT Department.
Software requirements	The Canvas Learning Management system can be run on most popular web browsers such as Internet Explorer, FireFox, Safari, etc. Canvas runs on Windows, Mac, Linux, iOS, Android, or any other device with a modern web browser. NOTE: Detailed information on installation and use will be provided in Canvas. Python is available from: http://www.python.org/download/ - **Available for both Mac and Windows**
	**Printing @ CSN:** Printing in CSN classrooms, computer labs and libraries falls under the new Print Wise initiative, designed to help save natural and fiscal resources. Print Wise provides each CSN student with a $10 credit toward printing at the start of each semester, which will provide for up to 200 black and white copies at 5¢ a page, or 40 color copies at 25¢ a page. After that, you may put money into your account online or at the CSN Cashier's Office to purchase additional prints at the same rate. It is **your responsibility** to maintain your printing accounts to cover printing expenses during each semester.
Advising & Coaching Services:	Advisor/Success Coaches help students assess academic strengths and limitations, learn academic success strategies, explore careers, declare a major, navigate the educational system, access campus and community resources, and connect to campus life. The department also manages the CSN Faculty E-Alert System assisting instructional staff by working with students on strategies and interventions that lead to successful course completion. Charleston Campus Bldg. D – Student Services Area 651-7367 North Las Vegas Campus 1100 Student Services Area 651-2626

	Henderson Campus Bldg. B – Room 120 651-3103
Counseling	Counselors assist students who are on: academic warning/probation/suspension and financial aid warning/suspension. Counselors prepare academic suspension and financial aid appeals. Counselors help students who are having academic challenges and also those who have been referred by their instructors through the MyCoyotePLAN Early Alert program. Counselors connect students to college and community resources and help them evaluate their options to make informed decisions. For more information about Counseling & Retention Services, please visit: https://www.csn.edu/counselingdepartment
Additional Notes	**My responsibilities:** 1. I will reply to your e-mail messages within one day. Replying to phone calls may take at most two days. 2. I will make sure to accommodate all your learning needs and will answer your questions in a timely manner. 3. I will try my best to resolve any issues. 4. I will return feedback and your grade on assignments within one week of the due date. **Your responsibilities:** 1. Stay active in classroom discussions and activities. Let me know if you find any discrepancies in the syllabus, course material, or activity due dates, as soon as possible. 2. Watch the deadlines for exams and ask questions. 3. Do the best you can in the class and don't hesitate to ask for help. 4. You will review my feedback on your assignments and will let me know of any questions or concerns as soon as possible.
Withdrawal Policy - IMPORTANT DATES	**IMPORTANT – I will NOT grant a W (withdrawal) once the official college deadline has passed. Please DO NOT ASK! Look at the section "Withdrawal Policy – IMPORTANT DATES for more information.**
CLASS POLICIES	All exams will be taken online administered through the Canvas system. Online sections - **You must take your exams at a physical campus location (West Charleston, Henderson, North Las Vegas campus sites).** Hybrid sections **– You will take your exam in class during the regular class hours.** All of the course material is available online in Canvas. While the design of this course allows flexibility in your scheduling, please realize that the deadlines are just as strict as any other course. You should check the calendar and discussion postings daily and allocate your time accordingly to complete the readings and to be prepared for the exams. Due dates will be strictly adhered to. You will use the mail and discussion features of Canvas to contact me or ask questions. **A note on the online/Hybrid environment:** In an effort to stay on task, I

	release chapter material and assignments on a timely basis. Exams can only be taken during the scheduled period. If you like to get these material earlier than the rest of the class, send me an e-mail message and I will give you access to this material. The links to assignments and exams will disappear after their respective due dates. Some students have complained that some links disappear from time to time. This is most likely due to system issues or incommutability with your web browser. Please send an e-mail message informing me about any links that have disappeared from your view. Online/hybrid sections offer flexibility where you need to manage your time to achieve success. I will assign programming projects that will be graded and you will have to take exams by specified due dates**. It is extremely simple to ignore due dates and fall behind which I hope we can seriously avoid!** **Hybrid section** – Using the hybrid format, you attend class once a week for 80 minutes and are expected to spend another 80 minutes online. Of course, in order to succeed, you need to study much more than the usual class time of 2 hours and 50 minutes a week. The rule is that you attend class about 3 hours a week and study a minimum of 6 hours a week outside of the classroom. Programming can be time-consuming. This is a single person class; meaning that just as in a normal classroom, you must turn in your own work. You are not allowed to collaborate or consult with anyone else while working on an exam. You are not allowed to collaborate on completing assignments. You should not be looking at each other's code for assignments. You can freely discuss items in the general sense. FAILURE TO ADHERE TO THIS POLICY WILL RESULT IN A ZERO FOR THE EXAM AND MAY LEAD TO REMOVAL FROM THE COURSE. Finally, Internet access is your responsibility. This class can be accessed from any computer with Internet access anywhere in the world. Therefore, excuses such as "My computer is not working" or "My provider was down" are not acceptable. If you find yourself in a real jam, you may drive to any CSN site and use one of the computers in the CSN open computer labs. Of course, if there is a computer problem originating from CSN, then I will take corrective action. But in all other instances, it is your responsibility to ensure your own Internet access.
Software Lab	The **software lab** may be available. See announcements in Canvas on the first day of class.
Centers for Academic Success (CAS)	Centers for Academic Success (CAS) provides quality DROP-IN academic assistance to all students enrolled in for-credit courses at CSN. Tutors are available for most general education courses and historically challenging courses. Academic learning support includes assistance with learning strategies, Canvas, Smarthinking online tutoring, Microsoft Office, reading, writing, oral presentations, math, and science. CAS tutors also provide support to study groups and assistance for placement test preparation in reading, writing, and math.

	CAS is open **Monday through Sunday** to be more accessible to all students. Hours for all locations are: Monday – Thursday 9:00 am to 6:00 pm and Friday – Sunday 11:00 am to 4:00 pm. You may visit www.csn.edu/centers-academic-success for more details on locations and hours. You may also contact us at one of our offices: Charleston Centers (7020-651-5732), North Las Vegas Learning Commons (702-651-4232), Henderson Learning Commons (702-651-3125).
EXCESS CREDIT FEE INFORMATION	Please visit https://www.csn.edu/excess-credit-information for information about the excess credit fee applied in certain circumstances.
TITLE IX Resources	More information is available at https://www.csn.edu/institutional-equity
U. Objectionable materials	This class will use a discussion forum. Please refrain from posting any objectionable or private information in these forums. If such information is posted, I will try to immediately delete your post. Failure to comply with this policy may also lead to dismissal from the class and referral to college administration for further actions.

We will have assignments due on a regular basis. The due date for each assignment will be announced in Canvas and all assignments are delivered and submitted through Canvas. You must be diligent in checking due dates for assignments.

Additional NOTES:

Exams can only be taken during the specified dates.

Assignments – 50%
Chapter quizzes – 10%
Midterm Exam – 20%
Final Exam – 20%

Here is a formula for how your final grade is calculated:
(The average of homework assignments * .5 + The average of chapter quizzes * .1 + midterm exam * 0.2 + final exam * 0.2)

So, if Joe's average on homework assignments is 80, he earns an average of 85 for chapter quizzes, 75 points on exam1, and 80 points on exam 2, what is his total score? What is his overall course grade?

Answer: (80 * .5 + 85 * .1 + 75*.2 + 80*.2) = 79.5
Letter grade: C+

The CANVAS system's grade book will allow you to track your grade for each item.

The following outlines the list of topics we cover in this course.

A complete list will be provided before the first class meeting.

Special NOTE: CSN academic integrity policy
Academic integrity is a legitimate concern for every member of the CSN college community. By joining the CSN college community, you accept the expectation to always take the ethical path and uphold the standards for integrity and honesty in your individual academic studies and to encourage others to do the same.
Stay out of trouble by following these rules:

Rule 1: You must not look at solutions or program codes that are not your own.
It is an act of plagiarism to submit work that is copied or derived from the work of others and submitted as your own. For example, using a solution from the Internet or a solution from another student (past or present) or some other source, in part or in whole, that is not your own work is a violation of the Academic Integrity Policy. Many infractions I see make use of solution code found online. The best way to steer clear of this possibility is not to search for online solutions to the programming assignments. Moreover, looking at someone else's solution code in order to determine how to solve the problem yourself is also an infraction of the Academic Integrity Policy. You should not be looking at someone else's code in order to solve the problems in this class. This is not an appropriate way to "check your work," "get a hint," or "see alternative approaches."
Rule 2: You must not share your solution code with other students.
You should not ask anyone to give you a copy of their code or, conversely, give your code to another student who asks you for it. Similarly, you should not discuss your algorithmic strategies to such an extent that you and your collaborators end up turning in the same code. Moreover, you are expected to take reasonable measures to maintain the privacy of your solutions. For example, you should not leave copies of your work on public computers nor post your solution code on a public website.
Rule 3: You must indicate on your submission any assistance you received.
If you received aid while producing your solution, you should indicate from whom you got help and what help you received. A proper citation should specifically identify the source (e.g., person's name, book title, website URL, etc.) and a clear indication of how this assistance influenced your work (be as specific as possible). For example, you might write "I discussed the approach used for sorting numbers in the sort_numbers function with Mary Smith." If you make use of such assistance without giving proper credit, you may be guilty of plagiarism.

It is also important to make sure that the assistance you receive consists of general advice that does not cross the boundary into having someone else write the actual code or show you their code. It is fine to discuss ideas and strategies, but you should be careful to write your programs on your own, as indicated in Rules 1 and 2
I have no desire to create a climate in which students feel as if they are under suspicion. The entire point of the Academic Integrity Policy is that we all benefit from working in an atmosphere of mutual trust. Students who deliberately take advantage of that trust, however, poison that atmosphere for everyone.

General Programming Assignment Requirements

1. For each program, you must work individually unless instructed otherwise. You may discuss the problem with classmates, but at no time should you discuss code in any form. *You may not show another student your code, share your file with another student, look at another student's code, or tell another student what to type.* **Evidence of academic dishonesty will result in a score of zero**

(see Academic Integrity section of syllabus). This applies to *all* students involved. If you're unsure about something, ask in advance.

2. Your program must adhere to the problem statement requirements and coding standards below. Violations will lead to deductions.
    a. A header comment *must* be included at the top of *each* submitted file. **Submissions without this header comment will receive a grade of zero.** The header comment must consist of the following information, including documentation tags (shown in bold face):

        /// **@author** Your name
        /// I pledge my word of honor that I have abided
        /// by the CSN Academic Integrity Policy while completing
        /// this assignment.
        /// **@file** The file name
        /// **@version** The date as YYYY-MM-DD
        /// **@brief** A brief description of the program (no more
        /// than one or two paragraphs)
        /// **@note** Time taken to develop, write, test and debug
        /// solution.

        Failure to disclose assistance will be interpreted as academic dishonesty.
    b. Basic blocks:
        i. Blocks will *always* use braces using methods demonstrated in class.
        ii. Statements in the block should be indented consistent with logical nesting. Use 4 spaces per indent level. Do not use tabs.
    c. Variables:
        i. Use descriptive names for variables using naming standards discussed in class.
        ii. Reduce the scope of variables so that they are only visible in the scope where they're used. Global *variables* are never permitted; global *constants* are.
        iii. Use one variable declaration for each variable you want to define (i.e., do not use the comma operator to declare multiple variables at one time). Variable declarations must appear at the beginning of the block of code in which they're used (i.e. do not intermix declarations with code).
        iv. Document the purpose of *every* variable.
    d. Statements:
        i. No more than one statement may be written on a single line.
        ii. The following may not be used: continue, goto, and break *not* in a switchstructure.
        iii. The use of exit should be reserved for unrecoverable errors only (e.g. failed memory allocation). Handle errors graciously wherever possible.
        iv. Lines of code should not extend beyond column 80.
        v. Diagnostic/debug print statements should be disabled/deleted in the final submission (i.e., submit a "shipping" version of your program). In advanced courses, use conditional compilation to enable/disable debugging statements.
    e. Functions:
        i. Use descriptive names for functions using naming standards discussed in class.

ii. All functions must be documented with the following information:
    1. Purpose – A statement or a set of statements that describes the purpose of the function.
    2. Pre-condition – A statement or set of statements that outlines a condition that should be true, or conditions that should be true, when called. The operation is not guaranteed to perform as it should unless the pre-conditions have been met.
    3. Post-condition – A statement or statements describing the condition that will be true when the operation has completed its task. If the operation is correct and the pre-condition(s) met, then the post- condition is guaranteed to be true.
    4. Parameter(s) – The purpose of each parameter should be described.
    5. Return value – For value-returning functions only, describe what the function returns.
    6. When using a single source code file, function documentation should be placed in comments directly above the function definition, not the prototype.
    7. When implementing a large project with multiple source code files, function documentation should be placed in comments directly above the function prototypes *in the interface file*, not in the implementation file.
iii. Function bodies should not be of extended length when easily separated into multiple functions (i.e., functions should do one thing and nothing more).
iv. Non-recursive functions should contain exactly one returnstatement.

3. Programs must be submitted on time. **Late programs *or* programs with syntax errors (i.e. do not compile due to errors) will receive a grade of zero**. Your program must compile cleanly (i.e. no warnings) and execute properly for credit. Note: It is better to submit a partially correct program *that compiles* than no program at all.
4. Submit the program file(s) electronically using the procedure shown on the problem statement requirements. You may submit your program file(s) as many times as you want before the deadline. *Each submission will replace any earlier submission*, so I can only see

and grade your most recent submission. Be sure to submit *all* required files with *each* submission. You cannot submit after the deadline (i.e., the drop box is closed).

Programming Grading Rubric

This document lays out common criteria used to grade programming assignments. Each criterion has several different levels of achievement, with a description of how a submission will attain that level and the number of points assigned for reaching it. Please email or ask me if you have any questions about this rubric.

Grading Standards

Every criterion will make up an approximate percentage of the grade given to a single programming problem as indicated in the "Approx. % of Grade" column. Points will be assigned for a criterion roughly along the lines of the guidelines of the "Excellent," "Above Average," "Average," "Below Average," and "Not Met" evaluations.

For example, a problem that was "Above Average" in the Program Specifications/Correctness criterion, "Average" in readability, and "Excellent" in all other areas would receive:
0.8*0.3 + 0.6*0.2 + 1*0.1 + 1*0.2 + 1*0.2 = 86% = B

		Excellent	**Above Average**	**Average**	**Below Average**	**Not Met**
Criterion	**Weight**	**100%**	**80%**	**60%**	**40%**	**0%**
Program Specifications / Correctness	40%	No errors. No warnings. Program always works correctly and meets the specification(s).	No warnings. Minor details of the program specification(s) are violated. Program functions incorrectly for some inputs.	One or more warnings. Program functions incorrectly for some inputs.	Significant details of the specification are violated. Program often exhibits incorrect behavior.	Program only functions correctly in very limited cases or not at all. Program does not compile and/or link.
Readability	20%	Code is clean, understandable, and well-organized.	Minor issues with consistent indentation, use of whitespace, variable naming, or general organization.	Multiple minor issues with consistent indentation, whitespace, variable naming, or general organization.	At least one major issue with indentation, whitespace, variable names, or organization.	Major problems with three or more of the readability subcategories.
		No errors. All variables	One or two places that could	Multiple places that could benefit	File header missing, complicated	No file header present. No

Documentation	20%	documented. All functions correctly documented. Code is well-commented. No spelling errors.	benefit from comments are missing them, or the code is overly commented. One or two variables not documented properly. One or two spelling errors.	from comments are missing them. More than two variables not documented properly. More than two spelling errors.	lines or sections of code not documented or lacking meaningful comments. More than four variables not documented. More than four spelling errors.	comments present. More than eight spelling errors.
Code Efficiency	10%	No errors. Code uses the best approach in every case.	Code uses poorly-chosen approach in one place.	Code uses poorly-chosen approaches in two places.	Code uses poorly-chosen approaches in three places.	Many things in the code could have been accomplished in an easier, faster, or otherwise better fashion.
Miscellaneous	10%	No errors.	One minor detail of the assignment specification is violated, such as incorrect filename.	More than one minor detail of the assignment specification is violated, such as incorrect filename.	Input/output varies significantly from that specified.	Significant details of the specification are violated, such as extra instructions ignored or entirely misunderstood.

* As a special case, if a program does not meet the specifications at all / is entirely incorrect, no credit will be received for the other criteria either.

Criteria

Program Specifications / Correctness

This is the most important criterion. A program must meet its specifications (whether from a textbook problem or as written in the assignment) and function correctly. This means that it behaves as desired, producing the correct output, for a variety of inputs. (In the beginning, I will be lenient with regards to producing correct output for all inputs, as we may not always have the tools needed to accomplish that, yet.) This criterion includes the need to meet specifications by writing a program in a specified way or using a required language feature, if such a thing is specified in the problem.

If a specification is ambiguous or unclear, you have two choices: You can either make a reasonable assumption about what is required, based on what makes the most sense to you, or you can ask the instructor. If you make an assumption about an ambiguous specification, you should mention that somewhere in a comment so that the reader/grader knows what you were thinking. Points may be taken off for poor assumptions, however.

Readability

Code needs to be readable to both you and a knowledgeable third party. This involves: Using indentation consistently (e.g., every function's body is indented to the same level)

- Adding whitespace (blank lines, spaces) where appropriate to help separate distinct parts of the code (e.g., space after commas in lists, blank lines between functions or between blocks of related lines within functions, etc.)
- Giving variables meaningful names. Variables named a, b, and c or foo, bar, and baz give the reader no information whatsoever about their purpose or what information they may hold. Names like principal, maximum, and counter are much more useful. Loop variables are a common exception to this idea, and loop variables named i, j, etc. are okay.
- The code should be well organized. Once we have learned about functions, code should be organized into functions so that blocks of code that need to be reused are contained within functions to enable that, and functions should have meaningful names. This is a concept that we will be learning about as we write more code, and so few points, if any, will be taken off for organization issues that we have not yet addressed.

Documentation

Every file containing code should start with a header comment. At the very least, this header should contain your name, the name of the file, and a description of what the included code does. Other details you might include are the date it was written, a more detailed description of the approach used in the code if it is complex or may be misunderstood, or references to resources that you used to help you write it.

All code should also be well-commented. This requires striking a balance between commenting everything, which adds a great deal of unneeded noise to the code, and commenting nothing, in which case the reader of the code (or you, when you come back to it later) has no assistance in understanding the more complex or less obvious sections of code. In general, aim to put a comment on any line of code that you might not understand yourself if you came back to it in a month without having thought about it in the interim. Like code organization, appropriate commenting is also something we will be learning about as we write code throughout the semester, so while corrections may be made, points will only be taken off for things that have been emphasized in class already.

Code Efficiency

There are often many ways to write a program that meets a specification, and several of them are often poor choices. They may be poor choices because they take many more lines of code (and thus your effort and time) than needed, or they may take much more of the computer's time to execute than needed. For example, a certain section of code can be executed ten times by copying and pasting it ten times in a row or by putting it in a simple for loop. The latter is far superior and greatly preferred, not only because it makes it faster to both write the code and read it later, but because it makes it easier for you to change and maintain.

Assignment Specifications

Assignments will usually contain specifications and/or requirements outside of the programming problems themselves. For example, the way you name your files to submit them to the course website will be specified in the assignment. Other instructions may be included as well, so please read the assignments carefully.