

Course Syllabus 1 - Information and Policies

This is an “emergency” doc, created in case the normal modules are not working, something that may happen if the hosting server for the page links goes down. Some of the links embedded in this page may not work, but if not, the content for those links is contained further down this document.

S.1 Course Description and Prerequisite

CS 3B is an in-depth study of essential intermediate *computer science concepts* and *OOP* programming techniques using the *Python* language. CS 3A is the *prerequisite* for this course.

Class inheritance, abstract classes, elementary data structures, Tkinter GUIs, NumPy arrays and multiple inheritance are among the many topics that will be covered in depth. Successful completion of CS 3B is required in order to continue with CS 3C, which is the study of algorithmic analysis and data structures, the centerpiece of all Python-based CS degree programs and vocations.

A working facility with simple algebra as well as good written English comprehension skills are both strong advisories.

S.2 Instructor

I am Michael Loceff, and you can email me at class@loceff.com. Typically you will ask questions through the private or public message center here in the course and only use email if you have trouble logging in.

S.3 Text and References

All of the important concepts will be covered in my modules. The text for the course is *recommended*, not required. It is *Python for Everyone, any Edition*, by Horstmann et. al./Wiley. You must have *some* reference, however, *you can use any Python textbook, new or used*, that fits your style and budget.

You can order this through the Foothill Bookstore at <http://books.foothill.edu/>, phone: (650) 949-7305.

S.4 IDE/Interpreter

You will need a (free) software package called a *IDE*, or **I**ntegrated **D**evelopment **E**nvironment along with a *Python Interpreter* (which includes a bytecode *Compiler*). Your instructor will recommend a preferred *IDE/Interpreter* product for your course. Two that run on both Windows PCs and Macs are **Python/IDLE** (or just *Idle*) and PyCharm. If you are facile on an *IDE* not

recommended by your instructor, you are welcome to use that, instead. However, assistance in the forums regarding *IDE/Interpreter* specifics will be limited to the recommended *IDE*.

S.5 Communication

Public Forums

Questions and comments should be posted to the Discussions Tool (DT) which you can reach by clicking on Discussions on the left menu. I will usually reply within a few hours. Unless a question is of a private nature (i.e. grades, registration issues), please use the public Discussions. Also, feel free to answer your fellow student questions even if you only have a guess as to what the answer is. It's great to engage in conversation with each other in this manner.

Steps needed to post your public questions and comments for this course can be found on the Canvas Discussions section, below.

First Week Required, Afterwards Recommended

No points are awarded for contributions, and there are *no weekly requirements*, but it's good collegial form to participate, inquire and assist. *Also, you must post an introduction* in the first week of class or you will be *dropped as a "no show"* according to the college requirements.

Do Not Post Homework Code

Whether you have a question or suggested answer, *never post exact homework code* to forums. Create a separate small program to display your issue or illustration.

Private Messages

Please use *public* DT for any question or comment that involves understanding the modules, tests or assignments. If you have a confidential question (grades or registration) use the Message Tool (MT) by first clicking on Inbox at the far left, then selecting this course and your intended recipient (usually me.)

Steps needed to post your public questions and comments for this course can be found on the Canvas Inbox Instructions, below.

Zoom Meetings



Your instructor may schedule *required* or *optional* virtual conferences using the **Zoom** conferencing tool. These may be one-on-one, or multi-student, depending on the purpose. If so, you will be asked to *sign-up* for a 15-30 minute session from the Calendar Tool on the *far left column* of your Canvas page. The *link for the meeting* will be supplied in an Announcement or Private Inbox Message.

Posting Program Code

You can post code to the public Discussions that is not directly from your Assignment. If you have an Assignment question, translate that into a piece of code that does not reveal your answer or submission, exactly.

When posting *code fragments* (i.e., portions of your program) into questions, make sure these code fragments are perfectly indented and that they are properly formatted. For details see the required resource module Posting Code int Questions section, below.

Do not post entire programs and ask "what's wrong?" or "is this good?" That's frivolous and indicates you have not tried to narrow down the problem. Find exactly what you want to know about and post only that part of the code.

STEM Success Center

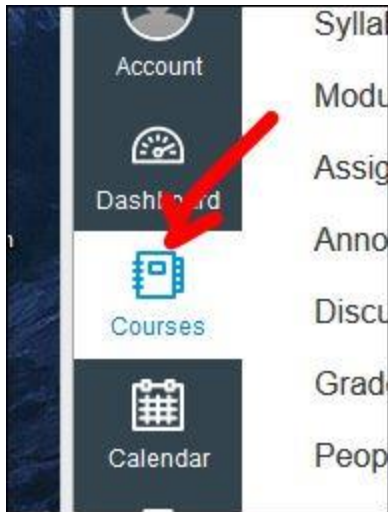
If the online forums here are not enough, please visit the [STEM Success Center page](#) and explore *Tutoring Schedules by Subject*, *Study Group Schedule*, and *STEM Center Weekly Workshop*, among many other useful links.

A **.pdf** of this quarter's *STEM center Flyer* can be found at the end of this document.

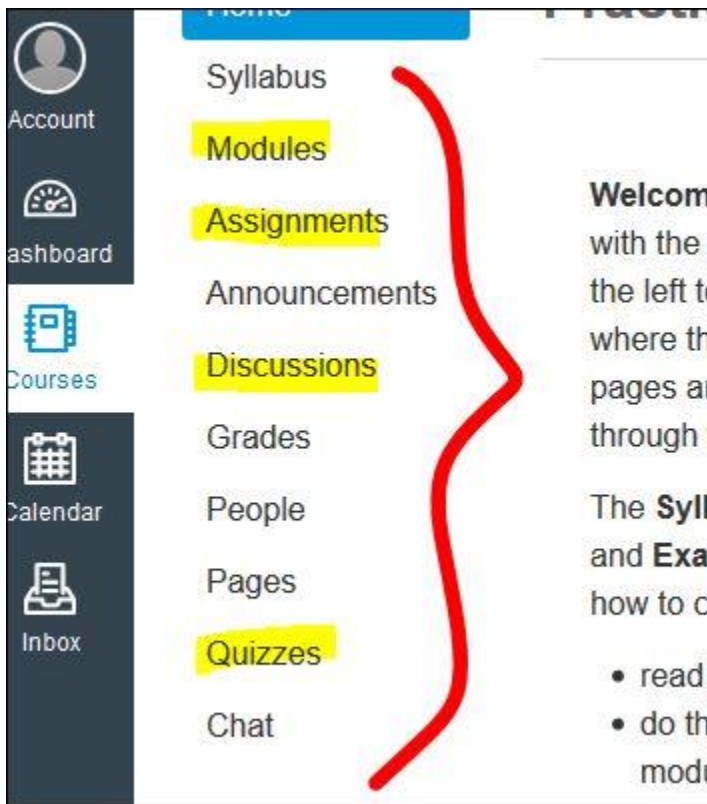
The STEM experts are qualified to help you with Assignments or Modules without giving you an answer that will short-circuit your discovery process. Let them know that you are not to receive actual Assignment solution code or even fragments. They probably know this already, but it's your responsibility to avoid submitting something that was written by a tutor or another person.

S.6 Where Everything Happens

Access the various areas of your course by first selecting this course through the **Canvas** Courses choice in the *far left* ...



... then examining our *course choices menu*, also on the left side of the screen, but slightly to the right of Courses:



- **Assignments:** submitted through the Assignments Tool (AST).
- **Tests:** taken through the Quizzes Tool (QT).
- **Questions or comments:** posted using the **Discussions Tool (DT)**.
- **Other areas:** You'll find the names self-explanatory, and you can investigate them on your own.

S.7 Grades

Your grades are based on

- Programming lab Assignments (nine @20 points each = 180 points, about **58%**),
- Quizzes (three @14 points each = 42 points, about **13%**), and
- Exams (midterm @30 points and final exam @60 points = 90 points, about **29%**)

for a total of **312** points (**100%**).

	Absolute Grading Scale	
	% needed for	this grade
97		A+
91		A
88		A-
86		B+
80		B
78		B-
75		C+
67		C
60		D
< 60		F

There will be opportunities for *extra credit bonus points*. These *may* be in the form of *advanced options* of some Assignments, a Bonus Assignment, or a private Zoom Meeting with me. The total possible **bonus points** will be about **50** points, or about **16%** of the final grade.

S.8 Drops and Withdrawal

For a complete reference of all withdrawal dates and deadlines refer to the Foothill College registration page at the college web site here:

[Foothill Fall 2018 Calendar](#)

To stay enrolled in this class, you must participate regularly in your lab assignments and exams. This is part of the *class participation* that online classes must possess in order to maintain their transferability and accreditation.

You will be dropped by me for any of the following:

- If you do not *post* an *introduction* to the "First Week Introductions" Discussion Forum by *Wednesday of the second week*, you will be dropped for non-participation.
- If you do not get **100%** on a short Course Syllabus and Assignment Policy Quiz by *Wednesday of the second week*, you will be dropped for non-participation.
- *Missing a scheduled quiz or exam* without prior notice will result in an automatic drop.

- If you *do not login* for **nine (9)** consecutive days I will drop you. (See exception below.)
- If you *receive a zero* on **any two (2)** Lab Assignments due to *non-submission*, I will drop you. (See exception below.)

A non-submission zero includes a submission that has *no merit* or shows *no serious design/coding/debugging* and is merely an attempt to avoid the drop.

Exception to Above Policies:

If the non-participation that has just been described occurs partially beyond the last date to drop, I may not be able to drop you, and you may receive whatever grade that your points dictate. Therefore don't assume that you can simply stop participating late in the quarter and you will be dropped. If you intend to drop please do so yourself, so you don't accidentally end up with an unintended "F."

If you decide to drop the class, please let me know. I cannot allow anyone who has dropped to continue to have access to the material.

S.9 Collaboration

Working together on homework = ZERO + Dean of Student's Office.

Husbands and wives, roommates, and friends taking the course together: don't discuss ungraded homework with each other outside the public forums. Instead, direct all of your questions to the public forums where everyone can comment and I can moderate the discussion. Do not look for answers on cheater web sites or pay-for-help web sites.



Any variation of collaborating or copying programming lab assignments is prohibited. The assignment must be 100% your own work. Changing a few variables around to make them look different won't fool me. And if it does fool me, you probably had to change so many things that you knew enough to do it yourself in the first place.

You can talk about the modules all day long off-line if you wish. This rule only applies to lab assignments. There is a place to ask for help with homework: the Public Discussions labeled for that purpose or the *STEM Success Center*. I will spend hours helping you each week, both individually, and in groups. You can even answer each other's questions in the Public Discussions. If I think you are giving too much information away, I'll edit your post. So there is no reason to ask your fiancée or your cousin's neighbor's lead guitarist.

If you accept help from someone who is not trained to teach without giving away the answer, it will short-circuit your learning process -- you will actually become weaker. Now, you don't have to agree with me - but you do have to follow the rule. If you want to take a class where you get to solve problems in groups, there are other sections with instructors who have that option. But

if you stay in *this* class, you are agreeing to do the lab assignments on your own or with help from us, here, in this course's public forum.

For those of you wishing to give help, please do not give away the answer. Either tell the person where they can look to find the solution, give them a general idea or ask them to ask me. Don't post actual assignment code.

S.10 How to Ask a Question

"There's no such thing as a bad question" is a myth. I don't know how the rumor got started.

It is easy to make sure your question is a good one: Make it specific. An example of a bad question is, *"My program doesn't work. Here it is. Would you please see if you can tell me what I am doing wrong? Gretel"* Gretel is lazy. An example of a good question is, *"My program doesn't work. Through trial and error I have determined that the problem lies in the following five lines, but I can't seem to narrow it down any further. Can you help? Hansel."* Hansel made an attempt to organize and isolate the problem prior to asking for help. When he gets my answer, he is sure to remember it because he is prepared to hear exactly what he needs to know.

Another example: BAD: *"I don't understand the assignment. I'm lost. Please help. Jack."* The reason this is a bad question is that there are a million things I might say to get Jack on the right track, but I can't know which ones to focus on because I don't know where Jack's misunderstanding lies. Jack hasn't given me any help to help him. GOOD: *I understand the homework description up until you say 'XYZ'. But I'm not sure what you mean by 'XYZ'. In the lectures 'XYZ' seems to be ... but here it seems to mean something different. From that point on, things get hazy because of this mismatch. Would you resolve this apparent difference for me? Jill."* Here, Jill has told me exactly the first point at which she is confused so I know what to tell her to set her straight.

I am not discouraging questions: I want you to ask. Through them, I get a chance to communicate with you. But narrow down the question. Show me you have tried to answer it and have made some progress. Show me exactly where you seem to be faltering so I can know how to help you. The same holds true if you are posing your question to a fellow student or to the whole class.

S.11 To Obtain Disability-Related Accommodations ...

... please contact **Disability Resource Center (DRC)** at the start of the quarter. To contact **DRC**, you may:

- Visit **DRC** in Building 5400
- Visit the [DRC Web Page](#)
- Email **DRC** at adaptivelearningdrc@foothill.edu
- Call **DRC** at 650-949-7017 to make an appointment

S.12 Expanded Content

- **Week 1** - Review of OOP, Idle, statics, enums, scope, text formatting, instance methods, the *self* reference, inner classes.
- **Week 2** - Review of lists, queues, sorting, recursion, binary search, early duck typing, NumPy arrays .
- **Week 3** - Binary, hex, bit operators, Cellular Automata, multi-dimensional arrays using lists and Numpy, stacks, multi-file projects.
- **Week 4** - Inheritance, method overriding, proto-base class object, constructor chaining with `super()`, more stacks.
- **Week 5** - Python GUI with Tkinter, input widgets, buttons, layouts, frames, event binding and handlers, listener classes, RPN calculator.
- **Week 6** - Deep copies, operator overloading, abstract classes, raising and trying python exceptions, error design, more duck typing.
- **Week 7** - Lambdas, class serialization, writing/reading JSON file formats, mathematical booleans, circuit and algorithmic complexity, seven segment displays, scope, namespaces and naming conflicts in inheritance.
- **Week 8** - More GUI, canvas widgets, sequences, mappings and containers, iterators, iterables, the linked list ADT (Abstract Data Type).
- **Week 9** - The general tree ADT, removal and recursion in trees, traversals, function objects.
- **Week 10** - Advanced Inheritance, multiple inheritance, method resolution order, `super()`'s magic attributes, `*args` and `*kwargs`.
- **Week 11** - Monotonicity, local precedence and the C3 method resolution order.
- **Week 12** - Final quiz.

You can access the official course outline of record for all **CS** courses here:

[Foothill College Catalog](#)

From that page, select **Dept: Computer Science** → **Search**, and from there, select any CS course whose official outline you want to review.

Course Syllabus 2 - Activity and Due Dates

This is the second of your two-page syllabus. The first page has the general policies and rules for the course.

S.14 Weekly Activities

Every week you have two lessons, or Modules, to study and one Lab Assignment to turn in. There are exceptions (see calendar, below), but this is the basic drill. This course is a lot of fun, and a lot of work. To pass it you have to make time to do both of these activities.



Weekly Time Estimate

- **Module Reading - about five hours.** This includes pasting code into your compiler and trying it out.
- **Lab Assignment - about six - 10 hours.** This varies greatly with individuals. Some students take one hour, some take 15 hours.

Typical Week

Here is the day-by-day breakdown of a typical week. Some weeks differ, but this will help you understand approximately what you are facing on a weekly basis.

Typical Week	
Monday (first 2 or 3 weeks only)	Read resource module R
Tuesday	Read module A
Wednesday	Assignment due (2 PM)
Friday	Read module B

S.15 Other Activities: *Discussions, Announcements, Tests*

Discussions

You can ask me or other students questions in the Discussion area. I hope you will be active in this area. Read through the recent Discussions posts every time you log in to make sure you gain the benefit of other students' questions.

Weekly Posts Recommended (Not Required)

Other than the *first week's introduction*, you are not required to post every week. However, if you are having difficulty, you should reach out and ask questions.

No Exact Homework Code Allowed

Please phrase questions in plain English or use non-homework code examples to demonstrate your question or suggested answer when posting.

Follow Module 3R When Posting

Code fragments must be formatted according to Module 3R to receive an answer. Otherwise, we'll ask you to fix the formatting and we'll check back to answer the question once the formatting is achieved.

You must also **post an introduction** by **Wednesday of the second week** (first 10 days of quarter) to avoid being **dropped as a no-show**.

Announcements

You will see an Announcement area in the **Canvas** course tools menu on the left. Check that area every time you login for late-breaking news. Announcements are a required part of the course and should be considered an **extension** of the Syllabus.

Tests

Exams

You have **two exams**,

- a **30-point** Midterm on **Friday** of the **sixth week**, and
- a **60-point** Final Exam on **Tuesday** of the **12th week**.

You are to take the Midterm in a single **one-hour** sitting and the Final Exam in a single **two-hour** sitting.

Quizzes

You have **three quizzes**, each worth **14 points**.

- A Course Policy Quiz will be due on **Wednesday** of the **second week**. You can re-take this up to 30 times until **Wednesday midnight**. It is a **14 question** quiz. Keep trying until you get **100%** (14/14). You must get 100% to stay enrolled in the course. It is basically a free, guaranteed 14 points unless you simply don't try or don't read the syllabus and assignment guidelines.
- Python Content Quiz #1 will be due on **Friday** of the **third week**. It is a **seven question** quiz.
- Python Content Quiz #1 will be due on **Friday** of the **eighth week**. It is a **seven question** quiz.

You are to take the quizzes in one **single 30 minute** sitting (except for the Course Policy Quiz, which can be taken multiple times, your highest score being the official one).

Time Window for Tests

All tests will be available *starting 6 PM the day before it is due* and remain open until *midnight the day it is due* (except for the Course Policy Quiz, which is open for *10 days before it is due*). There are no make-ups if you fail to take a quiz or exam. Failing to do so will result in an *automatic drop*.

If you exceed the time limit or hit the due date/hour (midnight), the test will be submitted automatically with the answers you have chosen up to that point. (However, since the Course Policy Quiz can be repeated, you may try again after each attempt.)

S.16 Official Calendar

Official Due Dates for Course

Date:	Day	Read Module	Lab Assignment Due 2 PM	Take Quiz/Test
Sep 24	Monday	Syllabus & Resource 1R		
Sep 25	Tuesday	Week 1A		
Sep 28	Friday	Week 1B		
Oct 1	Monday	Resource 2R		
Oct 2	Tuesday	Week 2A		
				Course Policy Quiz
Oct 3	Wednesday		Assignment 1	Last Day to Post Introduction
Oct 5	Friday	Week 2B		
Oct 8	Monday	Resource 3R		
Oct 9	Tuesday	Week 3A		
Oct 10	Wednesday		Assignment 2	
Oct 12	Friday	Week 3B		Python Quiz 1
Oct 16	Tuesday	Week 4A		
Oct 17	Wednesday		Assignment 3	
Oct 19	Friday	Week 4B		
Oct 23	Tuesday	Week 5A		
Oct 24	Wednesday		Assignment 4	
Oct 26	Friday	Week 5B		
Oct 30	Tuesday	Week 6A		
Oct 31	Wednesday		Assignment 5	
Nov 2	Friday	Week 6B		Midterm Exam
Nov 6	Tuesday	Week 7A		
Nov 7	Wednesday		Assignment 6	

Nov 9	Friday	Week 7B	
Nov 13	Tuesday	Week 8A	
Nov 14	Wednesday		Assignment 7
Nov 16	Friday	Week 8B	Python Quiz 2
Nov 20	Tuesday	Week 9A	
Nov 21	Wednesday		Assignment 8
Nov 23	Friday	Week 9B	
Nov 27	Tuesday	Week 10A	
Nov 28	Wednesday		Assignment 9
Nov 30	Friday	Week 10B	
Dec 4	Tuesday	Week 11A	
Dec 5	Wednesday		<i>Bonus Assignment</i> (No assignments accepted after 2 PM on this date.)
Dec 7	Friday	Week 11B	
Dec 11	Tuesday		Final Exam
Repeat			

No late Assignments accepted after **Dec 7, 2:00 PM**. Also, the Final Exam is not accepted late. It is due by midnight, Tuesday, **Dec 11**. You have three months to prepare for these deadlines.

Optional Modules

There is an occasional Module C in some weeks. I did not put this on the calendar. These Modules constitute optional reading meant for advanced and ambitious students.

S.17 Next Steps

Now that you have the idea, you can look up and see that in the first week you are supposed to read:

- Week 1, Monday — This Syllabus and Resource R1
- Week 1, Tuesday — Week 1A
- Week 1, Friday — Week 1B

Also, to avoid a *drop* for *non-participation* or *no/show*, be aware of some *Second Week deadlines*:

- Week 2, Monday — Read Resource R2 (Assignment Policy)
- Week 2, Wednesday, Latest — Get 100% on Syllabus and Assignment Policy Quiz, and
- Week 2, Wednesday, Latest — Post your *Introduction* to the Discussion Forums

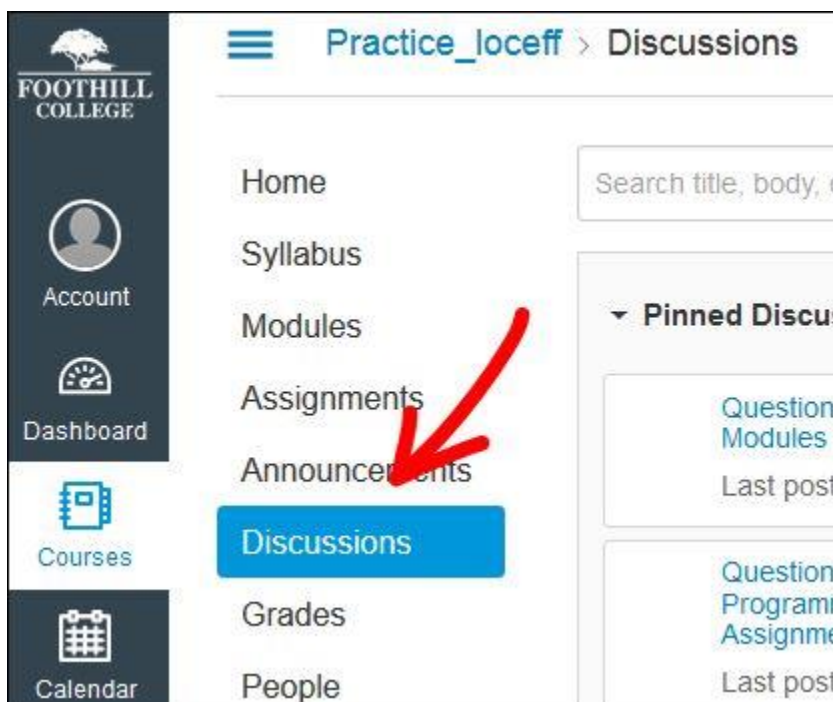
- (Week 2, Wednesday — Your first Lab Programming Assignment is due, but this will not result in a drop if you miss this deadline.)

Postings Questions or Comments to Canvas Discussions

This section addresses how and where to post simple questions to **Canvas Discussions** for our class. There are additional requirements when you want to include **code fragments** into your post. Those are not discussed on this page, but can be found in a separate handout [Pasting Code into Questions](#).

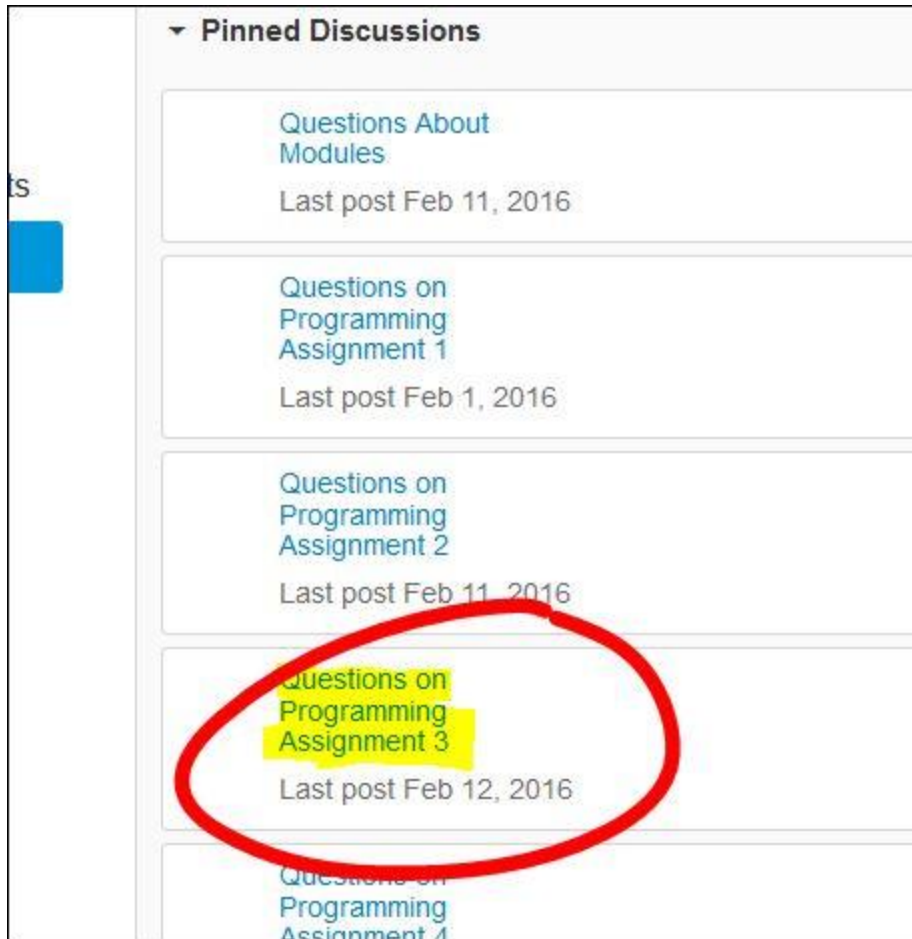
Posting in the Correct Discussion

Questions and comments should be posted to the Discussions Tool (DT) which you can reach by clicking on Discussions on the left menu.



I will usually reply within a few hours. Unless a question is of a private nature (i.e. grades, registration issues), please use the public Discussions. Also, feel free to answer your fellow students' questions even if you only have a guess as to what the answer is. It's great to engage in conversation with each other in this manner.

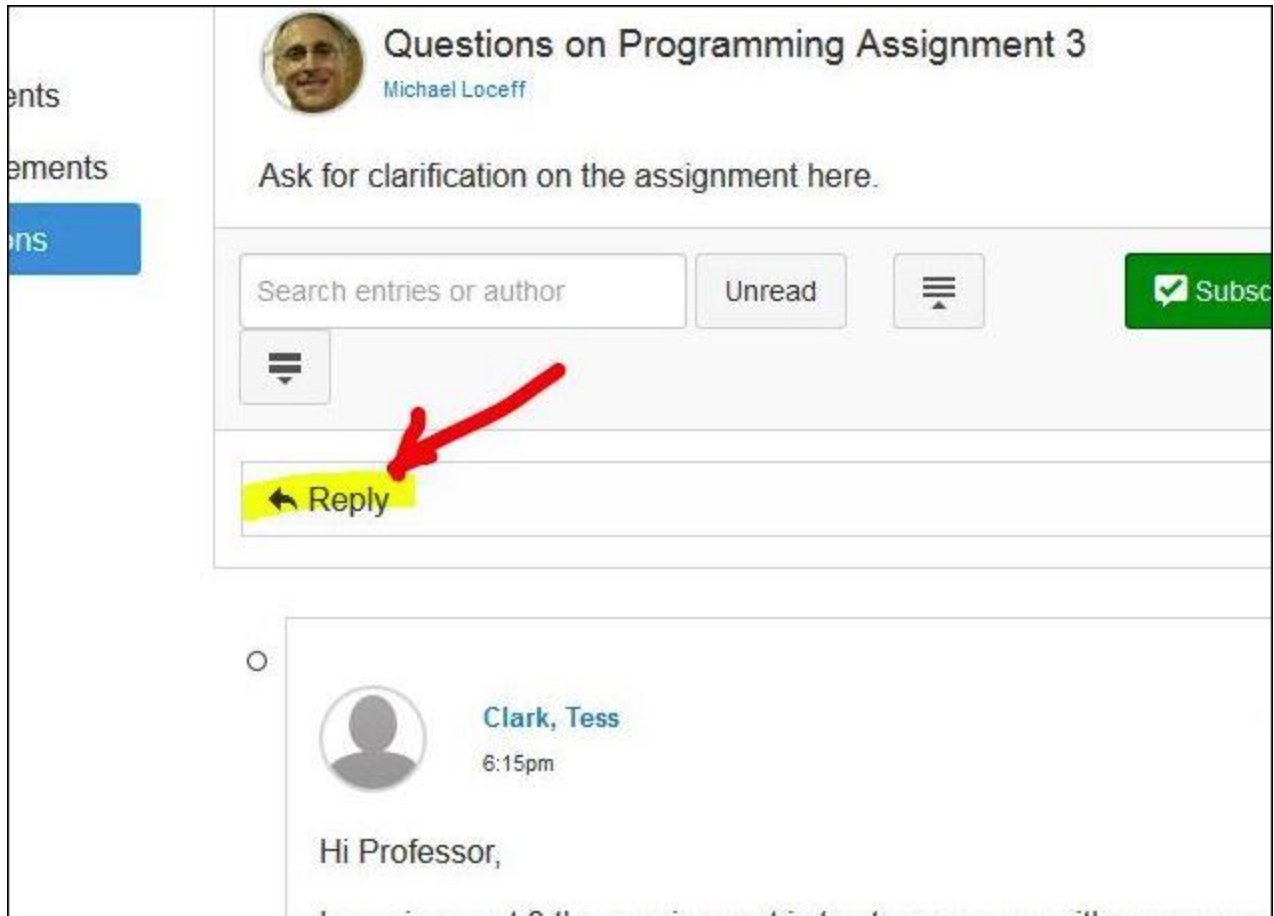
Discussion topics are already established, and you should **reply** to the appropriate one for your question or comment.



For example topics like *Questions on Assignment #3* already exist. If you have a new question about assignment #3, post it under *that* main "thread," as it's called. **Click the thread title** and you'll see a new screen devoted to that topic.

Starting a New Topic (Thread) Under the Main Thread

You cannot create a new major discussion category. However, you can start a new topic under an existing category. After you find the discussion that best matches your question, (*Assignment #3 Questions*, for example), *read all the existing posts to see if your question is answered* and/or fits better under those sub-topics. If it is truly a new question about *Assignment #3* (for example), Reply to my *original post* in that discussion:



Notice that the way to start this new question is to do so **at the very top** of the discussion thread. If yours is the first post in the discussion, there's only one place to reply and no confusion. However, if other students have already posted questions or other sub-topics to the discussions, you will have to ignore those posts, which appear below my original solicitation for questions. Instead, scroll to the top of that discussion. This will be the largest and top-most Reply icon (as the above screen shot demonstrates). Click on that and ask your question.

Adding a Follow-Up Question or Answer to a Posted Topic

If you want to continue a detailed topic already begun, find the **exact post** you want to address or augment. This will not be my original solicitation for questions but be found **further down the list of posts in that thread**. (Those are called **smaller threads** or **subthreads**.)

For example, you may want to answer one of your fellow students' questions or respond to my answer by asking for additional clarification. If so, **do not reply to the original post** at the top of the discussion, but **scroll down and find the subthread** post to which you want to respond. On each such post there's a smaller Reply icon that you can use to add your updated question or comment.

S

ents

Michael Loceff

Ask for clarification on the assignment here..

Search entries or author

Unread

Subsc

Reply

Clark, Tess

6:15pm

Hi Professor,

In assignment 3 the requirement is to stop program with error mess at certain condition. Did you mean that I should stop program and g user an Error message of my own (e.g.. "You entered improper num of...")?

Thanks,

Clark

Reply

Postings Private Questions

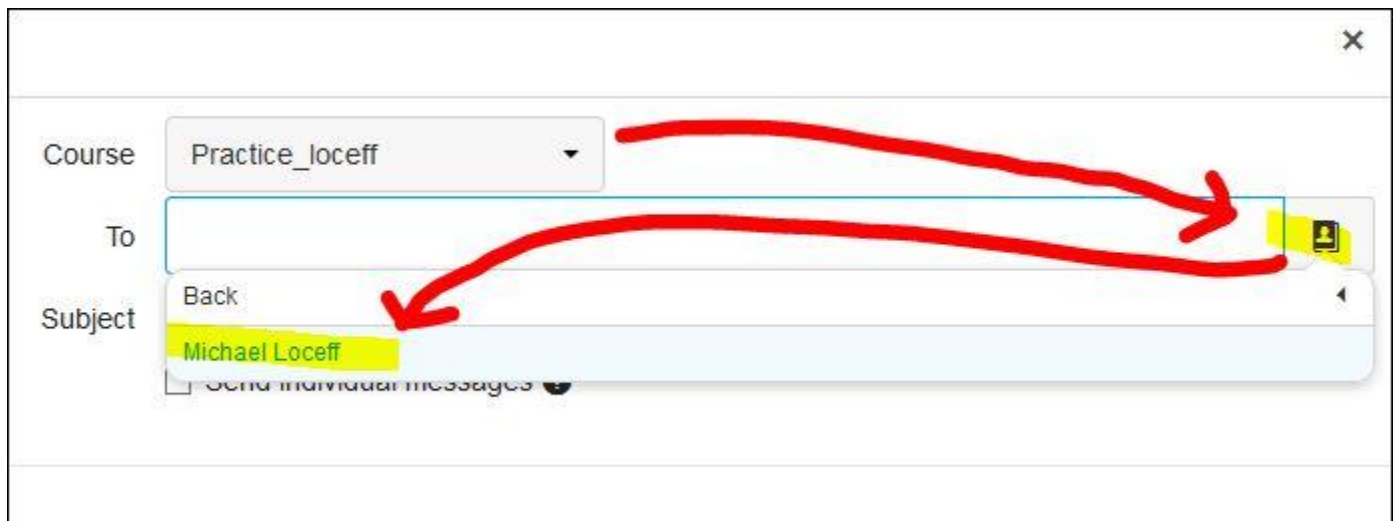
Canvas private messaging system **does not support** the embedded **Preformatted** text (or any type of formatting for that matter), therefore you must not type or paste programming code into your messages. Rather, state your short private message in plain English and **supply properly formatted code fragments as attachments**.

Private Messages

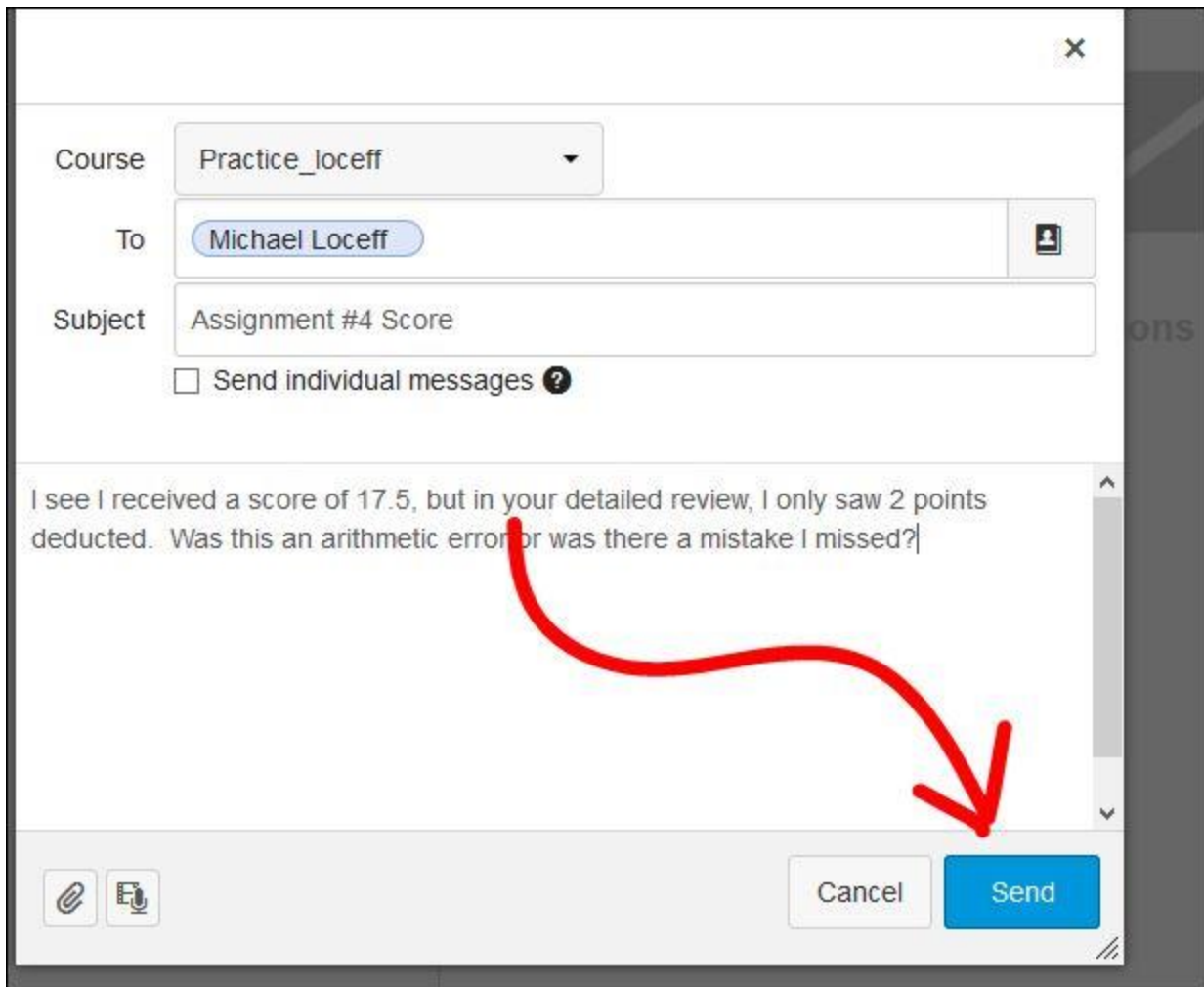
Please use *public* Discussions for any question or comment that involves understanding the modules, tests or assignments. If you have a confidential question (grades or registration) use the Message Tool (MT) by first clicking on Inbox at the far left, then clicking on the ***compose message icon:***



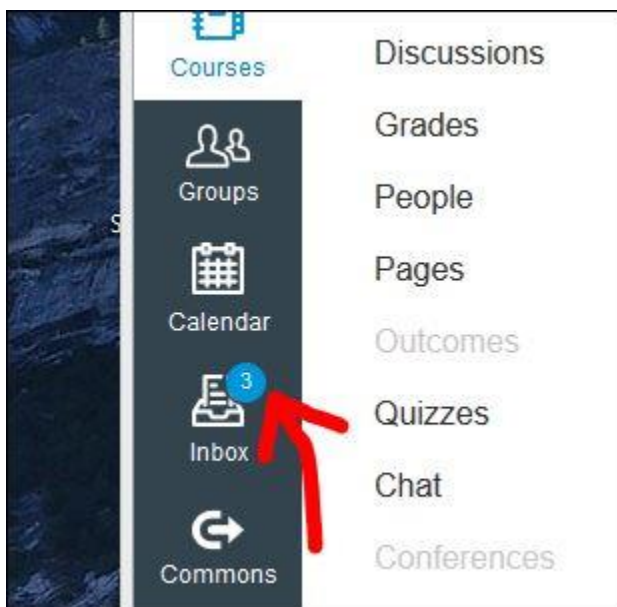
Next, choose this course and click the ***recipients*** icon which will take you to me or another recipient:



Compose a short private message and send it.



When it's been answered, you'll see 1 or more unread messages waitin in your Inbox.



Click into the Inbox to find and read the message.

Pasting Code Into Questions (Canvas)

This page only addresses the pasting of source code fragments into your *public* Discussions posts.

It does not apply to Canvas private messaging Inbox system; **Canvas does not support private message preformatted pasting.** See Syllabus (above) and Inbox references (below for more about private messages.

The Code Examples come from Java, C++ and/or Python. This page is shared for all languages.

CF.1. Canvas Buggy Discussion Tool

Canvas's Discussion Tool has *bugs* that cause it to make pasting code into discussions problematic. Also, the Discussion Tool version changes, so some bugs get fixed and others are introduced. This document describes how to work around the bugs that are present in the Fall 2018 version of the Discussion Tool.

CF.2 Try Firefox

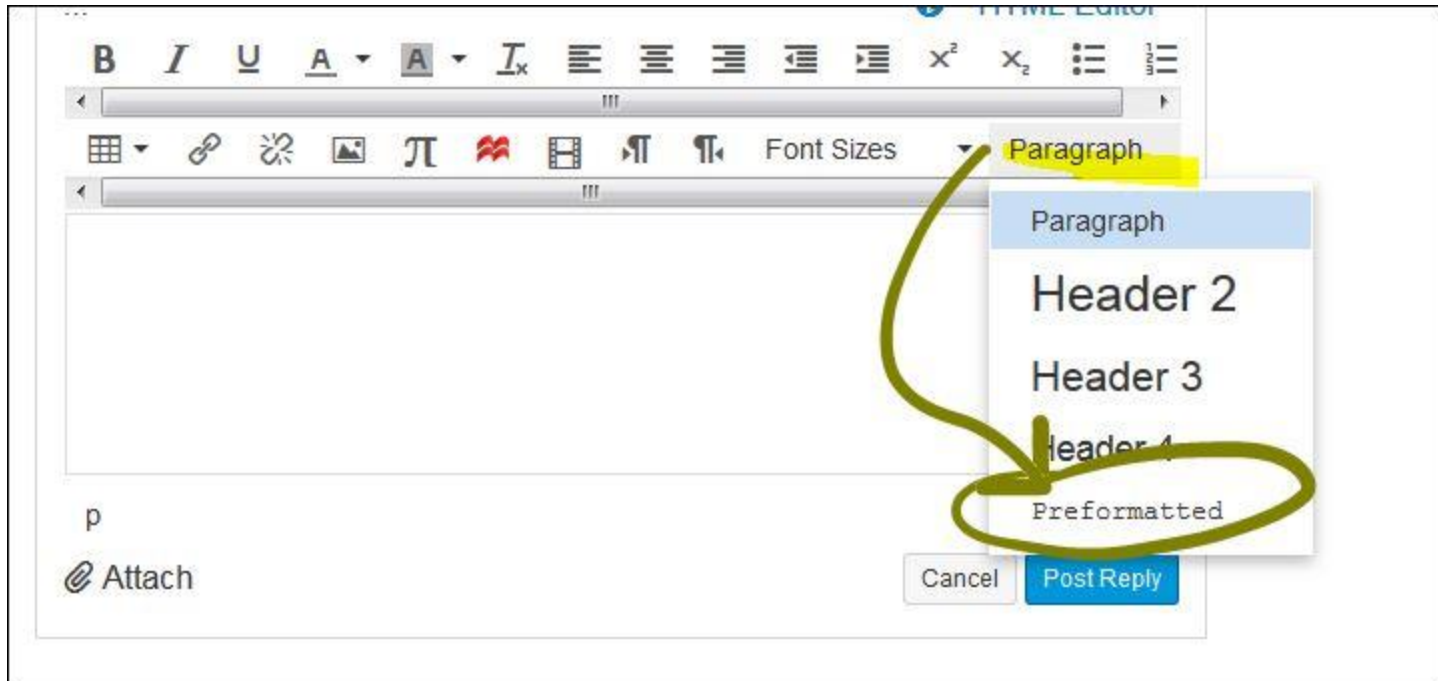
Since this is a programming class, you will have occasion to post *code fragments* - snippets of your program or of a program in the lecture. The most consistent browser for posting to Discussions is *Firefox*, so if you are having trouble with posts, try *Firefox*.

Note

If your code is not correctly formatted, you will be asked to reformat it before I will read the details and issue an answer.

CF.3 Paragraph vs. Preformatted

When posting code, it is essential that you use the **Preformatted** option in the drop-down format menu at the *top right* of your posting window. It is normally set to Paragraph (sometimes called "*Normal*") so we must change it when the time comes (pictured below). *However, there is much more to do than just this one move, so keep reading.)*



CF.4 Code Without Blank Lines (PC, Mac and Linux)

This section only works if you want to paste code fragments that do not have blank lines, like:

```
#include <iostream>
#include <ctime>
#include <string>
using namespace std;
// class Card prototype -----
class Card
{
public:
    enum Suit { clubs, diamonds, hearts, spades };
```

It won't work if you insert a blank line, like:

```
#include <iostream>
#include <ctime>
#include <string>
using namespace std;

// class Card prototype -----
class Card
{
public:
    enum Suit { clubs, diamonds, hearts, spades };
```

Your questions will typically be ordinary English, using Paragraph (*Normal*) formatting. However, if you want to show me some code, you will need both Paragraph text (to describe your problem in English) as well as Preformatted text (to show a few lines of code.)

Start your question using Paragraph text, then — when it is time to insert your code fragment — you will have to switch to Preformatted style. This is a little trickier than it sounds, but the following steps will guide you.

1. Make sure the original code fragment that you want to post is perfectly indented in its original location and has all tabs replaced by appropriate spaces so I and others can read it easily. *It is considered rude to ask for help on a program and give incorrectly indented samples.*
2. Start a post or reply to someone else's post. You will be given an *edit window*. Grab the lower-right triangle of the edit window to create enough room for both your **Normal (Paragraph)** text and the **Preformatted** code fragment.

Ask for clarification on the assignment here.

Search entries or author Unread

↓

↓

B *I* U A **A** *I*_x x^2 x_2 \sqrt{x} 12pt

<

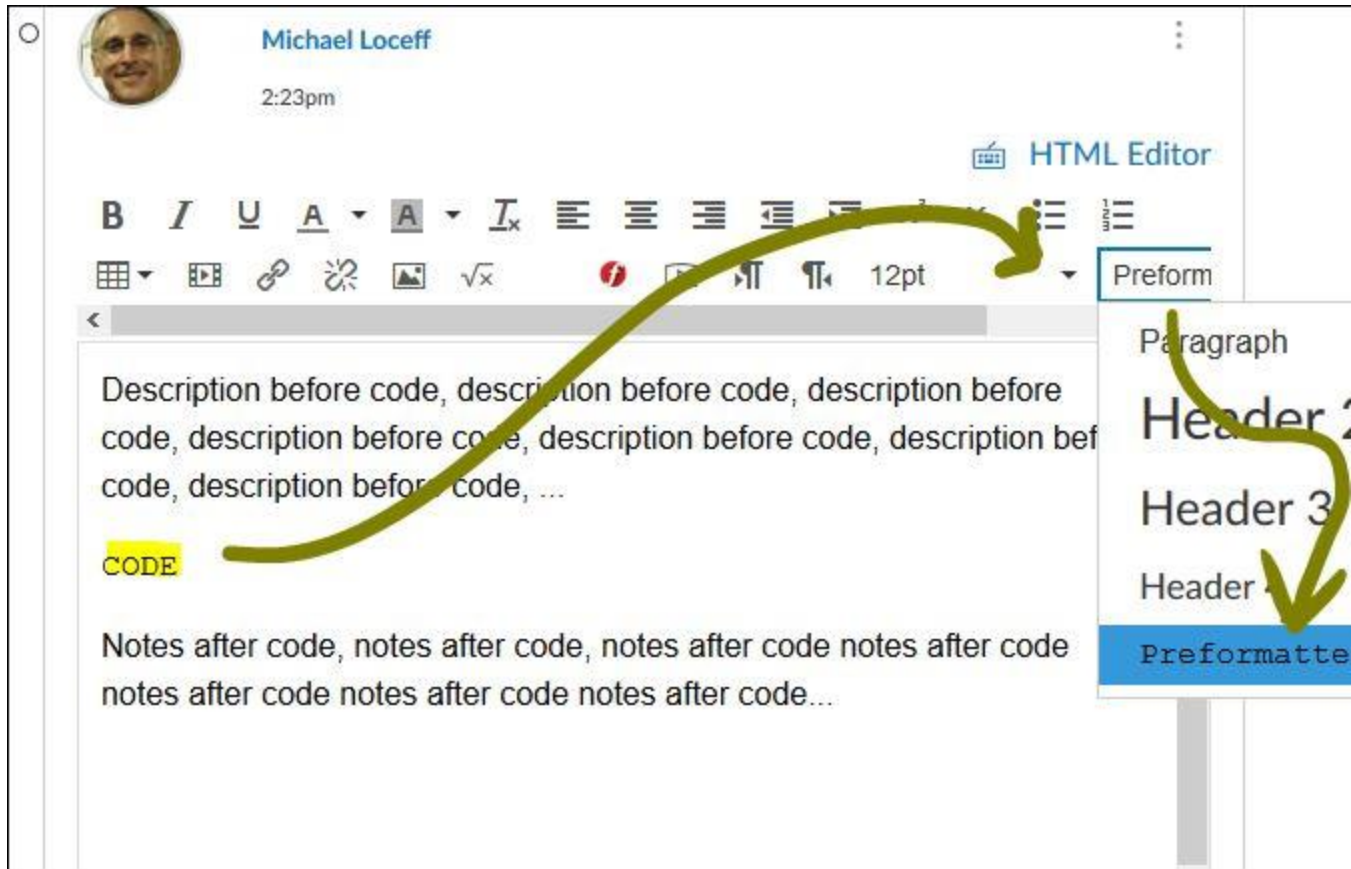
p

Attach Cancel

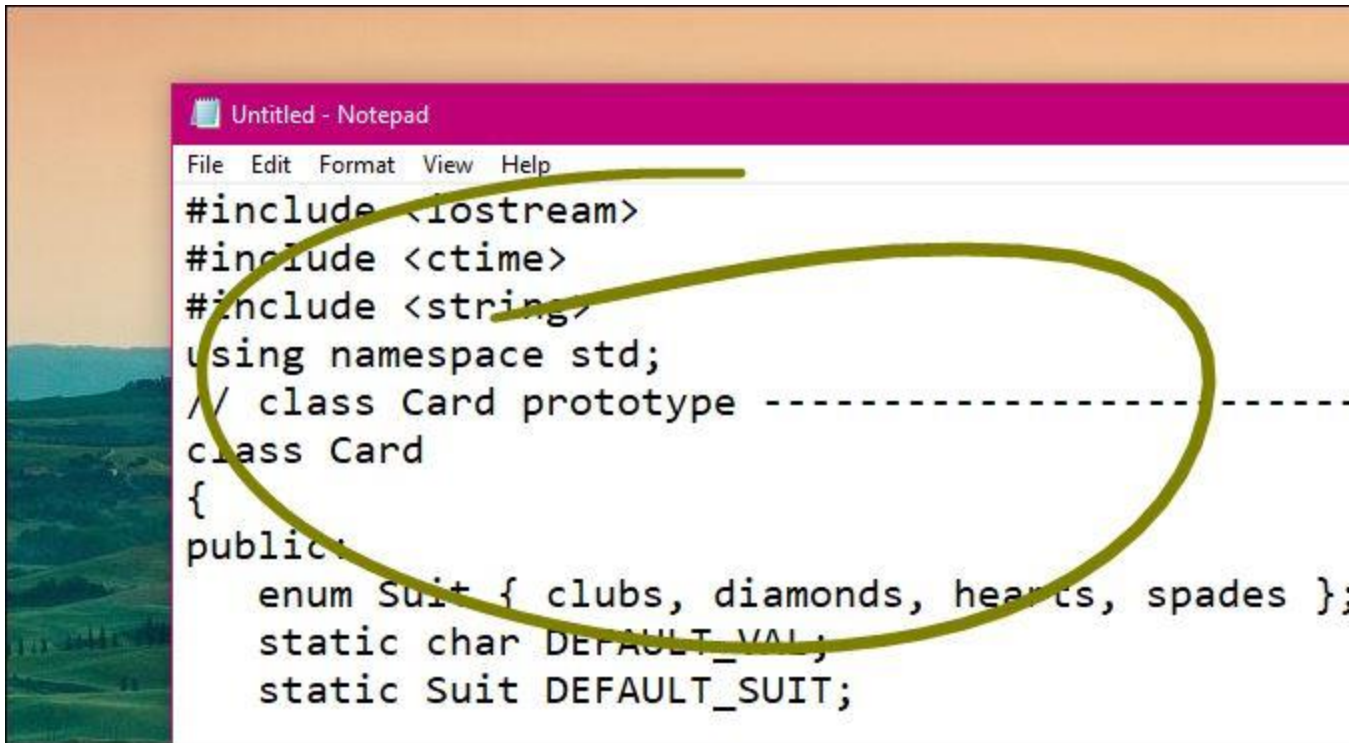
3. Write the plain English description of your question and, when you are ready to show code, do not type or paste the code. Instead, hit the **ENTER** key to give yourself a new paragraph, type a short word like "**CODE**" where the code will eventually go and, finally, hit the **ENTER** key again to add any extra plain English comments or questions after the code. Your screen will look something like this:

The screenshot shows an HTML Editor window. At the top left is a profile picture of Michael Loceff and the name 'Michael Loceff' with a timestamp of '2:23pm'. The editor's title bar says 'HTML Editor'. Below the title bar is a toolbar with various icons for text formatting (bold, italic, underline, text color, background color, bulleted list, numbered list, indent, outdent, link, unlink, image, link, font size) and a paragraph style dropdown set to 'Paragraph'. The main text area contains several lines of placeholder text: 'Description before code, description before code, description before code, description before code, description before code, description before code, ...'. The word 'CODE' is highlighted in yellow. Below this is another line of placeholder text: 'Notes after code, notes after code, notes after code notes after code notes after code notes after code...'. Handwritten green annotations include the word 'code' at the top right, 'will go' in the middle, and a checkmark pointing to the highlighted 'CODE'.

4. Select the word **CODE** and change the style from **Paragraph** to **Preformatted**. You will see a subtle difference in the look of the word (**CODE** → `CODE`):



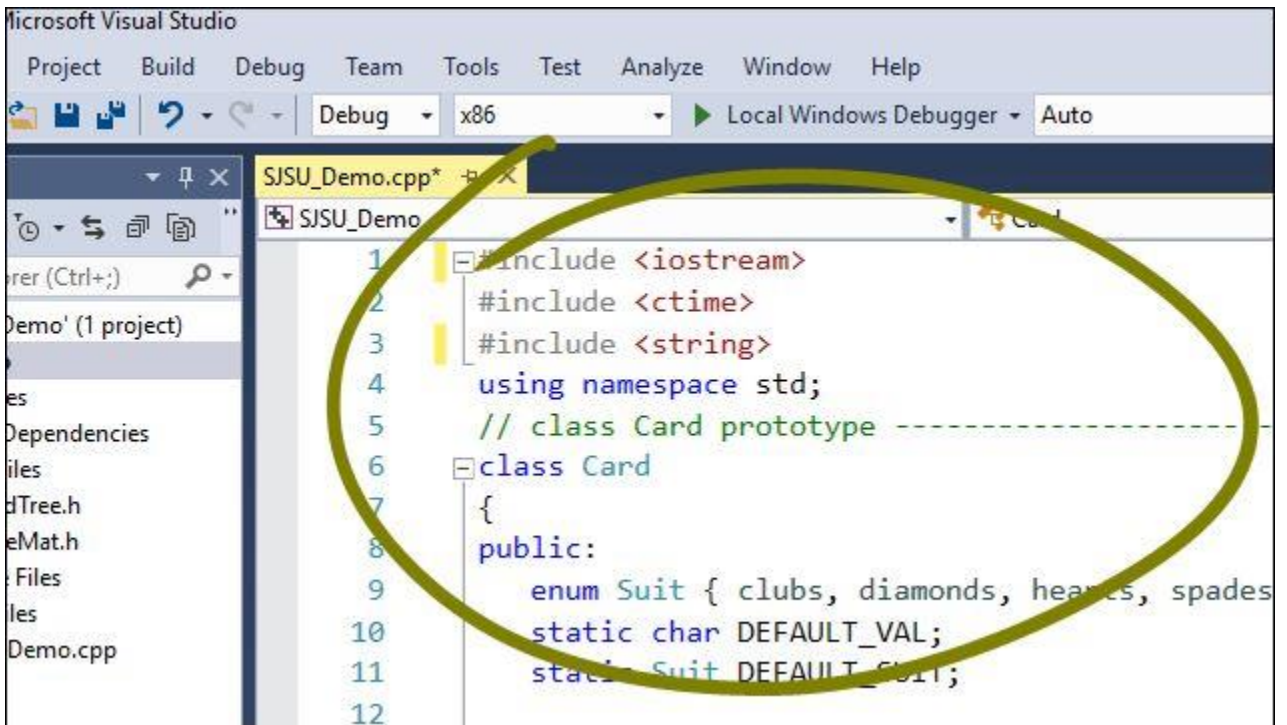
5. Find the programming fragment from your compiler *IDE*, *Text Editor* or a Modules lecture. Be sure it is perfectly formatted before you get it. Then copy it from the source file (**Control-C** from Windows, **Command-C** from Mac). Here is a sample copied from the Notepad *Text Editor*:



Untitled - Notepad

```
File Edit Format View Help
#include <iostream>
#include <ctime>
#include <string>
using namespace std;
// class Card prototype -----
class Card
{
public:
    enum Suit { clubs, diamonds, hearts, spades };
    static char DEFAULT_VAL;
    static Suit DEFAULT_SUIT;
```

and here's one copied from an *IDE's* text editor:

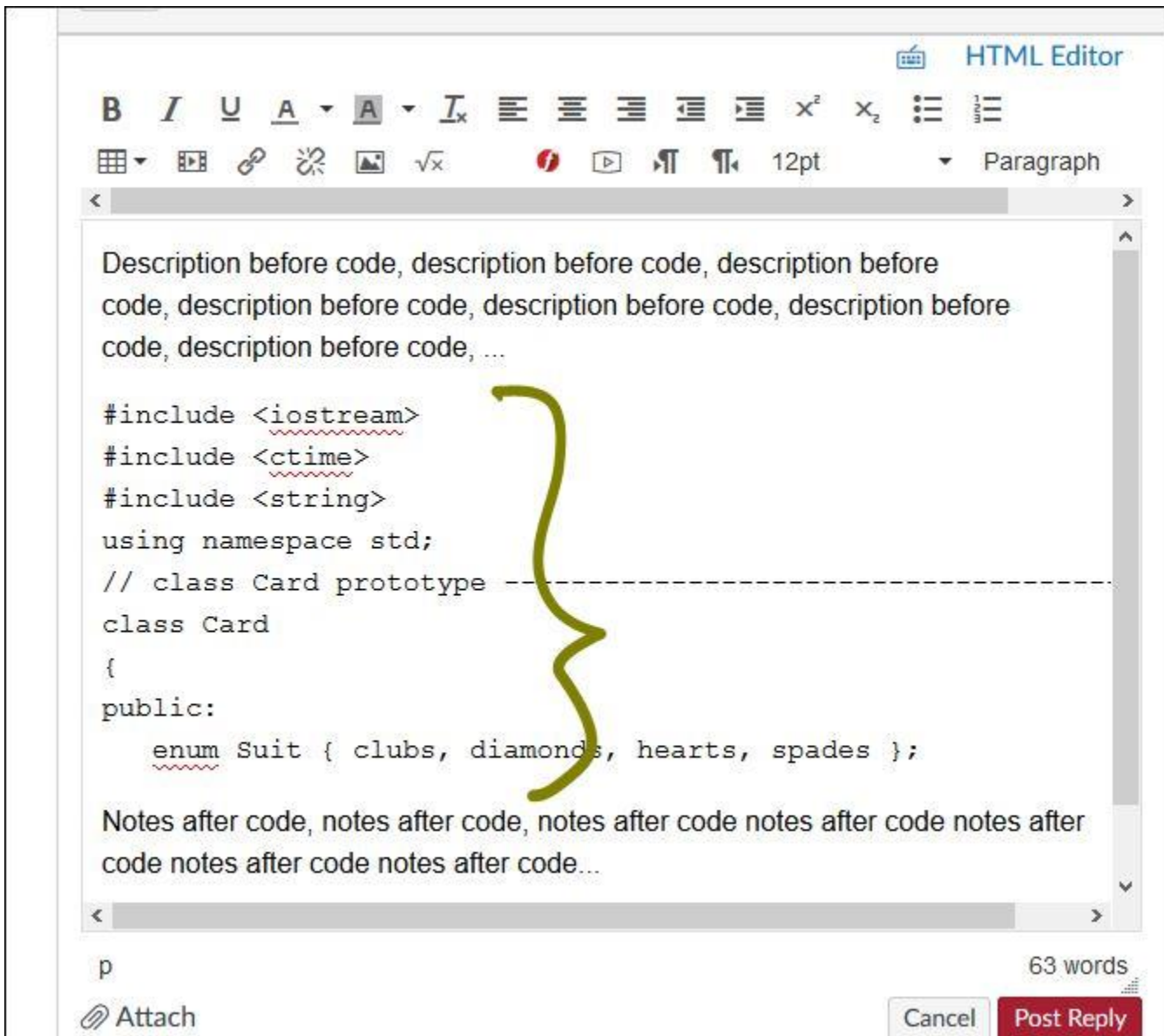


Microsoft Visual Studio

```
Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger Auto
SJSU_Demo.cpp*
SJSU_Demo
1 #include <iostream>
2 #include <ctime>
3 #include <string>
4 using namespace std;
5 // class Card prototype -----
6 class Card
7 {
8 public:
9     enum Suit { clubs, diamonds, hearts, spades };
10    static char DEFAULT_VAL;
11    static Suit DEFAULT_SUIT;
```

(The Code Example is from C++, but the same procedure for posting code applies to C++ and Python. This page is shared by all languages.)

6. Get back into your Canvas Discussions *edit window* and select just the word "CODE" (or whatever word you used). Paste your fragment using the keyboard shortcut **Control-V** for Windows or **Command-V** for Mac.




The screenshot shows a Canvas HTML Editor interface. The top toolbar includes various text formatting options like bold, italic, underline, and text color. Below the toolbar, the editor content area contains several paragraphs of placeholder text: "Description before code, description before code, description before code, description before code, description before code, description before code, ...". In the center, there is a code block for a C++ class named 'Card'. The code includes headers for <iostream>, <ctime>, and <string>, uses the 'std' namespace, and defines a class 'Card' with a public enum 'Suit' containing 'clubs', 'diamonds', 'hearts', and 'spades'. A large green curly bracket is drawn over the code block, indicating it has been selected. Below the code block, there is more placeholder text: "Notes after code, notes after code, notes after code notes after code notes after code notes after code notes after code...". At the bottom of the editor, there is a status bar showing "p" and "63 words", along with "Attach", "Cancel", and "Post Reply" buttons.

```
#include <iostream>
#include <ctime>
#include <string>
using namespace std;
// class Card prototype -----
class Card
{
public:
    enum Suit { clubs, diamonds, hearts, spades };
}
```

7. Scroll up and down to confirm the pasted code is in the correct font and the style is exactly as it was in your source IDE or module. Then click **Post Reply** or **Done**. You will see the question with the embedded code fragment:

Reply



[Michael Loceff](#) 2:23pm

Description before code, description before code, description before code, description before code, description before code, description before code, description before code, ...

```
#include <iostream>
#include <ctime>
#include <string>
using namespace std;
// class Card prototype -----
class Card
{
public:
    enum Suit { clubs, diamonds, hearts, spades };
```

Notes after code, notes after code, notes after code notes after code notes after code notes after code notes after code...

← Reply

8. If it doesn't look perfect, go back and edit it (upper right gear section):

Michael Loceff
2:23pm

Description before code, description before code, description before code, description before code, description before code, description before code, description before code, ...

```
#include <iostream>
#include <ctime>
#include <string>
using namespace std;
// class Card prototype -----
class Card
{
public:
    enum Suit { clubs, diamonds, hearts, spades };
```

That's it. Just wait for your instructor or classmates to answer the question.

CF.5 Code Containing Blank Lines (PC, Mac and Linux)

The steps above won't work if your code contains blank lines, like this:

```
/* CS 1A Lab 2 */

public class Foothill
{
    public static void main(String[] args)
    {
        int resultInt;
        double resultDub;
        int studID, numLet;

        studID = 22222222;
        numLet = 5;
```

. Here's what you'll see after you paste that code into the CODE section you created:

Michael Loceff
2:23pm

HTML Editor

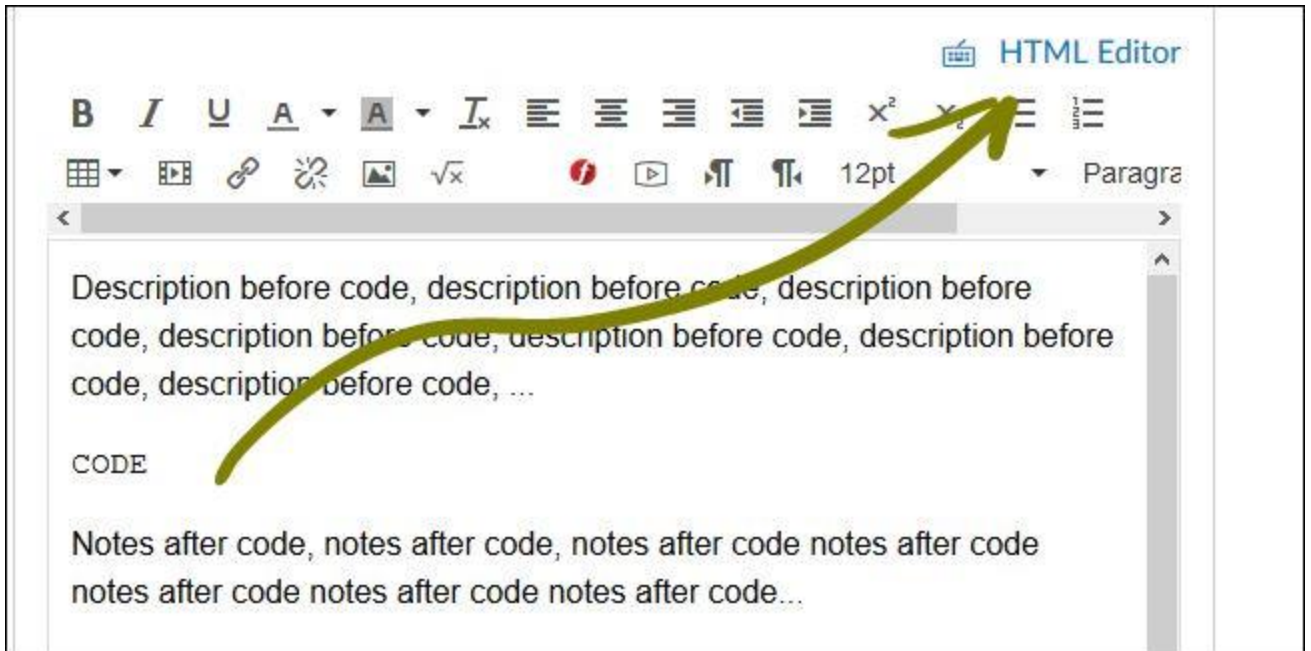
B *I* U A **A** *I* x^2 x_2 \sqrt{x} 12pt Paragraph

```
/* CS 1A Lab 2 */  
  
public class Foothill  
{  
    public static void main(String[] args)  
    {  
        int resultInt;  
        double resultDub;  
        int studID, numLet;  
  
        studID = 22222222;  
        numLet = 5;  
    }  
}
```

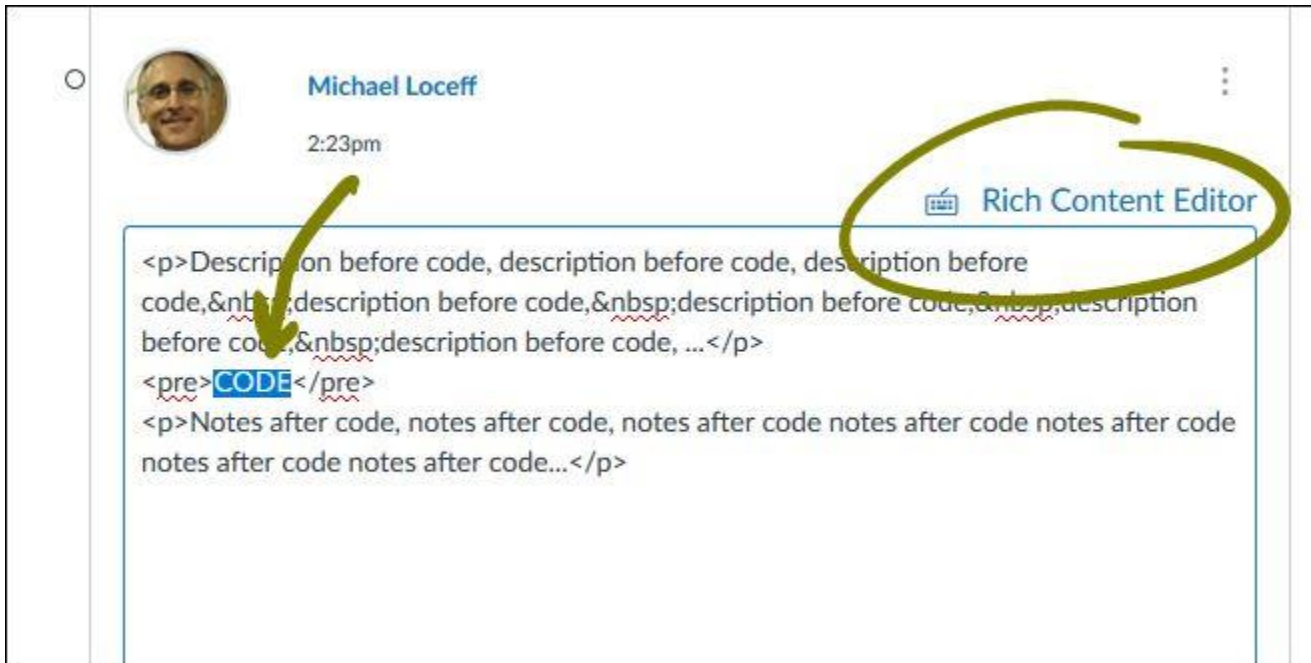
p 68 words

Done

So, after you have set your placeholder word CODE to **Preformatted**, find the **HTML Editor** selector and click it:



You'll get an uglified version of the post, with HTML tags. Find the word **CODE** between the tags `<pre>` and `</pre>` and select it (*don't select the tags*).



Now, replace the word `CODE` with your copied *code fragment*:




It still looks pretty ugly, but that's because you haven't returned to your **Rich Content Editor** yet, so do that by clicking **Rich Content Editor** selector and you'll get:

The image shows a screenshot of a code editor window. The editor has a toolbar at the top with icons for grid, zoom, link, undo, redo, and font size (12pt). The code is as follows:

```
Description before code, description before code, description before  
code, description before code, description before code, description befo  
code, description before code, ...  
  
/* CS 1A Lab 2 */  
  
public class Foothill  
{  
    public static void main(String[] args)  
    {  
        int resultInt;  
        double resultDub;  
        int studID, numLet;  
  
        studID = 22222222;  
        numLet = 5;  
  
Notes after code, notes after code, notes after code notes after code  
notes after code notes after code notes after code...  
p 68
```

A large, handwritten green curly brace is drawn on the right side of the code, encompassing the entire class body from the opening curly brace to the closing curly brace. The word "good" is written in a cursive, handwritten style in green ink to the right of the brace.

Now, post and it will look good:

 **Michael Loceff**
2:23pm

Description before code, description before code, description before code, description before code, description before code, description before code, description before code, ...

```
/* CS 1A Lab 2 */  
  
public class Foothill  
{  
    public static void main(String[] args)  
    {  
        int resultInt;  
        double resultDub;  
        int studID, numLet;  
  
        studID = 22222222;  
        numLet = 5;  
    }  
}
```

Notes after code, notes after code, notes after code notes after code notes after code notes after code notes after code...

Edited by [Michael Loceff](#) on Sep 21 at 3pm

← Reply

CF.6 Loss of Words in Angled Brackets (Second Method)

The problem with the second technique that lets you paste code containing blank lines is that even *it* won't work all the time. If you have words in *angled brackets* like `<iostream>` or `<sstream>`, as in ...

```
cs_2a_assignment_soln_9old.txt - Notepad
File Edit Format View Help
/* CS 2A Lab 9
 * Instructor Solution
 */

#include <iostream>
#include <string>
#include <sstream>
using namespace std;

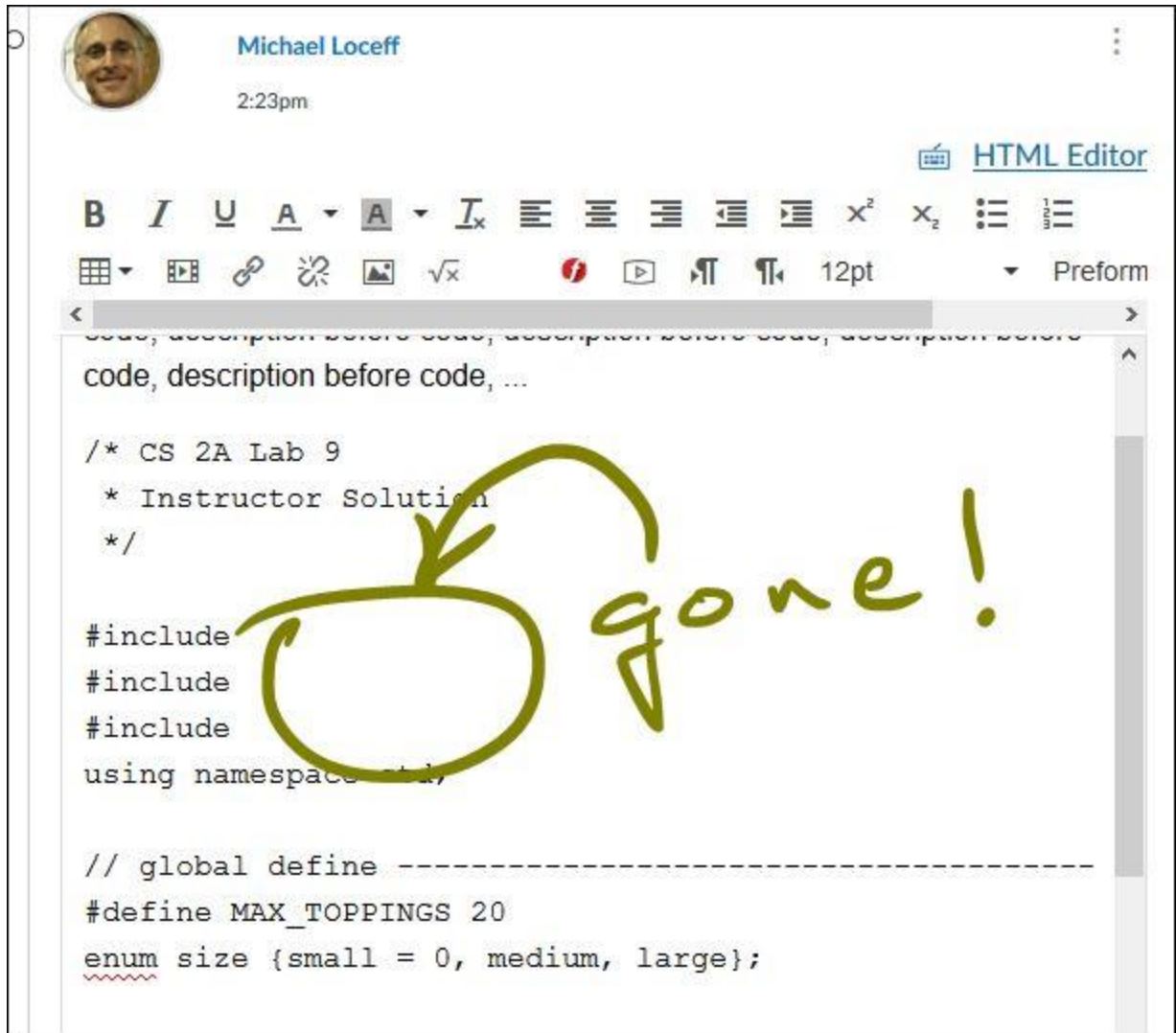
// global define -----
#define MAX_TOPPING 20
enum size {small = 0, medium, large};

// global-scope method prototypes
void displayMainMenu();
char getSizeFromUser();

// class prototype -----
class PizzaOrder
{
public:
    static const string TOPPINGS_OFFERED[];
    static const double TOPPINGS_BASE_COST;
    static const double BASE_PRICE;

    PizzaOrder( int size = small);
```

... those words will *disappear* after the process.



The rest of the code will be there, and the blank lines will be preserved, but those few angled-bracket words will be gone. However, this is usually something that you won't need to post in a question.

Summary:

If you have a *small fragment* with *no blank lines*, you can use the first process, and you won't lose these words. If you have a long fragment with blank lines, you'll have to accept losing these special bracketed words.

CF.7 Steps for Chrome

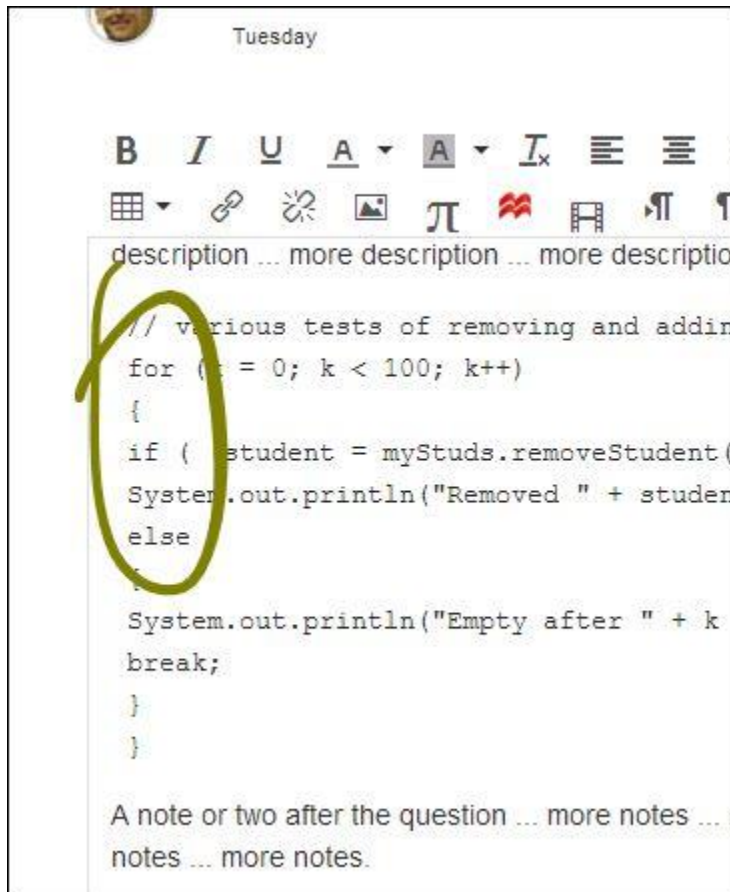
In Chrome, you can apply the first technique to code fragments with or without blank lines, but after you paste, you'll notice that the *indentation is lost*. We have to fix it. Here are the details

6. Get back into your **Canvas** Discussions *edit window* and select just the word "CODE" (or whatever word you used). Paste your fragment according to

Windows → the keyboard shortcut **Control-V**, or right click "*Paste as plain test.*"

Mac→ the keyboard shortcut **Command-Option-Shift-V** (*not* Command-V), or right click "*Paste and match style.*"

Once you do the paste, notice how the code has the correct font and width, but everything is horribly pushed to the left edge. Indentation is lost:

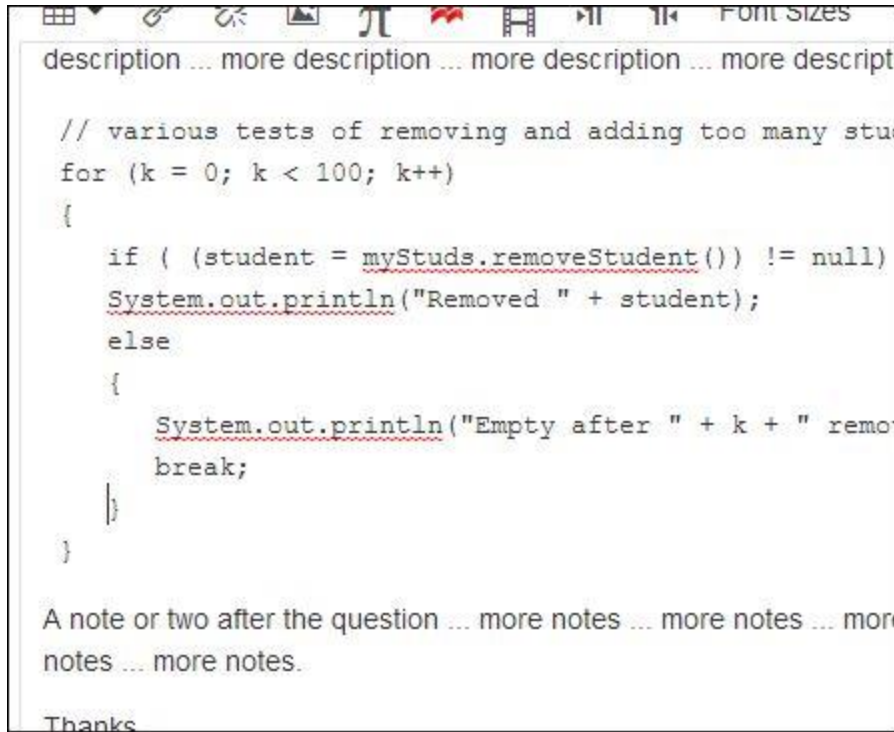


The screenshot shows a text editor interface with a toolbar at the top containing icons for bold, italic, underline, text color, background color, strikethrough, bulleted list, and numbered list. Below the toolbar is a text input area containing the following code snippet:

```
description ... more description ... more description
// various tests of removing and addin
for (k = 0; k < 100; k++)
{
  if ( student = myStuds.removeStudent(
System.out.println("Removed " + studen
else
}
System.out.println("Empty after " + k
break;
}
}
```

A green circle highlights the first few lines of code, showing that the indentation has been lost and the code is aligned to the left margin.

Place your cursor at the start of each line that needs indentation and manually enter the correct number of spaces to make that line — and the eventually the entire fragmet — perfect. This could take some time:

A screenshot of a web browser window. The top toolbar shows various icons and the text "Font Sizes". The main content area contains a code snippet in a monospaced font. The code is as follows:

```
description ... more description ... more description ... more descript

// various tests of removing and adding too many stu
for (k = 0; k < 100; k++)
{
    if ( (student = myStuds.removeStudent()) != null)
        System.out.println("Removed " + student);
    else
    {
        System.out.println("Empty after " + k + " remo
        break;
    }
}

A note or two after the question ... more notes ... more notes ... more
notes ... more notes.

Thanks
```

Of course, I have tried to convince you that using *Firefox* will do all this without manual work, but if you insist on using *Chrome*, you have the steps. If you can get these results using *Safari*, *IE* or *Edge*, fine. I haven't found a way to do that, yet, but you may.

Note

Do not try to type code fragments directly into the course web site's discussion window. Always prepare your code fragment first in your compiler or Notepad and copy/paste from there into the web site's discussion window.

STEM Success Center

Get help before
you need it!

The STEM Center offers a place to study on your own or with friends. Computers, printers, scanners, coffee and tea. Feel free to bring your lunch...even heat it up in our microwave! And, we offer free printing to STEM students.

We also offer workshops in Chemistry and Computer Science, and even online Computer Science tutoring.

Build-a-Workshop! Just get a study group of students together, and one of our instructors can join your group weekly.

We'll even help you form your study group.

WHERE: Room 4213

STEM CENTER HOURS:

Monday - Thursday:

7:30 AM - 9:00 PM

Fridays:

7:30 AM - 4:00 PM

Saturdays:

10:00 AM - 8:00 PM

NOTE: Tutors are available starting at 9AM Monday-Friday

ONLINE CS HELP:

Sunday: Noon - Midnight

Mon - Thurs: 8PM - Midnight

foothill.edu/stemcenter/onlinecs

**Math - Chemistry - Physics - Computer Science
Astronomy - Biology - Engineering - Accounting -
Economics**

Foundations Lab - Room 4201

While all students are welcome to use the whole STEM center, we've got a foundations lab just for students in MATH 48ABC, MATH 10, MATH 180, MATH 217/57,

CHEM 30A/30B, and ENGL 209/110.

Computers, software and dedicated, patient instructors just for you.

Monday-Thursday 9AM-8PM, Fridays 9AM-4PM and Saturdays 11AM-5PM

Accounting and Economics Lab - Room 4202

Students in all accounting and economics classes are invited to our lab.

Experienced instructors are there to help you:

Monday through Thursday 11AM - 5PM, Fridays 11AM - 2PM

Computer Science Lab - Room 4204

Find tutors familiar with your CS assignments in Java, Python, C++ and more.
foothill.edu/stemcenter **FHSTEMcenter**