



Academic Integrity At MIT

What is Academic Integrity?

Consequences for Academic Dishonesty

Responsible and Ethical Conduct of Research

About the Handbook

Citing Your Sources

What is Plagiarism?

Why and What to Cite

Common Knowledge

Electronic Sources

Academic Writing

Writing Original Work

Avoiding Plagiarism

Quoting

Paraphrasing

Quote vs. Paraphrase

Summarizing

Good Note Taking

Writing Code

Collaboration

Cheating

Working Under Pressure

Helping You Succeed

Academic Integrity Handbook

Writing Code

Writing code is similar to academic writing in that **when you use or adapt code developed by someone else as part of your project, you must cite your source**. However, instead of quoting or paraphrasing a source, you include an inline comment in the code. These comments not only ensure you are giving proper credit, but help with code understanding and debugging.

When should I cite a source in my code?

- When you copy code from an external source. Whether you are copying a snippet of code or an entire module, you should credit the source.

- When you copy the code and adapt it, you should still credit the source. You were not the original developer of the code.

How should I cite the code?

- Generally, the URL and the date of retrieval are sufficient. Add more details if it will help the reader get a clearer understanding of the source.

- If you adapted the code, you should indicate "Adapted from:" or "Based on" so it is understood that you modified the code.

- Your instructor may have specific instructions on how you should or should not cite your sources. If you are not clear on what is acceptable, ask your instructor.

Use of Open Source Software

When you use code from an open source project, you need both to attribute the source and **follow the terms of any open source license that applies to the code you are using**. Keep in mind:

- When you download the source, the license is typically part of the download.

- Also, the source code itself will typically contain the copyright and terms of use.

- When you incorporate open-source-licensed code into a program, it is good practice to duplicate the copyright in your code, and/or store the license in a file with the code.**

- If you don't obtain the license with the download, you should be able to find it on the site of the open source project, such as [Apache HTTP Server site](#), or on the [Open Source Initiative \(OSI\) site](#).

Instructors determine the specific expectations around re-use of code in each class.

Often, the requirements are described in the collaboration policy for the class. If policy is not clearly described in the course materials and you are not sure what is acceptable, ask your instructor.

Collaboration policy from Spring 2012 6.005 Elements of Software Construction: (used with the permission of Professor Rob Miller, Dept of Electrical Engineering & Computer Science)

We encourage you to help each other with work in this class, but there are limits to what you can do, to ensure that everybody has a good individual learning experience.

Individual work

Problem sets in this class are intended to be primarily individual efforts. You are encouraged to discuss approaches with other students but your code and your write-up must be your own.

You may not use materials produced as course work by other students, whether in this term or previous terms, nor may you provide work for other students to use.

It's good to help other students. But as a general rule, during the time that you are helping another student, your own solution should not be visible, either to you or to them. Make a habit of closing your laptop while you're helping.

During code review, you will see classmates' solutions to a problem set. While it is fine to take inspiration from their approach, do not copy their work.

It's fine to use material from external sources like StackOverflow, but only with proper attribution, and only if the assignment allows it. In particular, if the assignment says "implement X," then you must create your own X, not reuse one from an external source.

It's also fine to use any code provided by this semester's 6.031 staff (in class, readings, or problem sets), without need for attribution. Staff-provided code may not be publicly shared without permission, however, as discussed later in this document.

Example 1

- Alyssa and Ben sit next to each other with their laptops while working on a problem set. They talk in general terms about different approaches to doing the problem set. They draw diagrams on the whiteboard. When Alyssa discovers a useful class in the Java library, she mentions it to Ben. When Ben finds a StackOverflow answer that helps, he sends the URL to Alyssa. OK.
- As they type lines of code, they speak the code aloud to the other person, to make sure both people have the right code. INAPPROPRIATE.
- In a tricky part of the problem set, Alyssa and Ben look at each other's screens and compare them so that they can get their code right. INAPPROPRIATE.

Example 2

- Jerry already finished the problem set, but his friend Ben is now struggling with a nasty bug. Jerry sits next to Ben, looks at his code, and helps him debug. OK.
- Jerry opens his own laptop, finds his solution to the problem set, and refers to it while he's helping Ben correct his code. INAPPROPRIATE.

Example 3

- Louis had three problem sets and two quizzes this week, was away from campus for several days for a track meet, and then got sick. He's already taken two slack days on the deadline and has made almost no progress on the problem set. Ben feels sorry for Louis and wants to help, so he sits down with Louis and talks with him about how to do the problem set while Louis is working on it. Ben already handed in his own solution, but he doesn't open his own laptop to look at it while he's helping Louis. OK.
- Ben opens his laptop and reads his own code while he's helping Louis. INAPPROPRIATE.
- Ben has by now spent a couple hours with Louis, and Louis still needs help, but Ben really needs to get back to his own work. He puts his code in a Dropbox and shares it with Louis, after Louis promises only to look at it when he really has to. INAPPROPRIATE.

Example 4

- John and Ellen both worked on their problem sets separately. They exchange their test cases with each other to check their work. INAPPROPRIATE. Test cases are part of the material for the problem set, and part of the learning experience of the course. You are copying if you use somebody else's test cases, even if temporarily.

Note that in the examples marked inappropriate above, both people are held responsible for the violation in academic honesty. Copying work, or knowingly making work available for copying, in contravention of this policy is a serious offense that may incur reduced grades, failing the course, and disciplinary action. Copying, or helping somebody copy, may result in an F on your transcript that you will not be able to drop.

This policy applies to all coursework that is handed in by an individual: problem sets, reading exercises, nanoquiz makeups, etc.

Group work

You should collaborate with your partners on all aspects of group project work and in-class collaborative exercises, and each of you is expected to contribute a roughly equal share to design and implementation.

You may reuse designs, ideas and code from your own work earlier in the semester (even if it was done with a different partner). You may also use any code provided by this semester's 6.031 staff.

You may also use material from external sources, so long as: (1) the material is available to all students in the class; (2) you give proper attribution; and (3) the assignment itself allows it. In particular, if the assignment says "implement X," then you must create your own X, not reuse someone else's. Finally, your group may not reuse designs, ideas, or code created by another group, in this semester or previous semesters.

Although it is common practice to adapt code examples found on the web,

- You should never copy code from other students.** Your peers are *not* considered an authorized source.

- You should not simply re-use code as the solution to an assignment.** Like academic writing, your code can incorporate the ideas of others but should reflect your original approach to the problem.

Examples of citing code sources:

Example 1

In describing the class *PluginProxyUtil* in the Apache Project source code, the developer cites the source as a post in a forum and includes the URL, author and date:

```
/**
 * A utility class that gives applets the ability to detect proxy host settings.
 * This was adapted from a post from Chris Forster on 20030227 to a Sun Java
 * forum here:
 * http://forum.java.sun.com/thread.jspa?threadID=364342&start=120
 * [...]
 */
```

(Source: Apache Project source code <http://svn.apache.org> retrieved in July 2019.)

Example 2

In the function *OutputTraceToStream* in the Google Chrome *stack_trace_win* source code, the developer cites the source code as the Microsoft Developer Network and includes a URL:

```
// Code adapted from MSDN example:
// http://msdn.microsoft.com/en-us/library/ms680578\(VS.85\).aspx
```

(Source: https://github.com/adobe/chromium/blob/master/base/debug/stack_trace_win.cc retrieved in July 2019.)

Example of open-source-licensed code:

At the top of the Google Chrome *stack_trace_win* source file, note the copyright and reference to the open source license:

```
// Copyright (c) 2011 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.
```

If you incorporate this code into a program, you should follow the terms outlined in The Chromium Authors' open source license file, which is shown below. While this license only requires that you duplicate the copyright and license if you are redistributing the code, **it is good practice to always duplicate the copyright in your code, and/or store the license in a file with the code**. This way, if you want to redistribute the code later, intellectual property reviewing becomes much easier.

```
// Copyright (c) 2014 The Chromium Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//
// Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//
// Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//
// Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
//
```

(Source: The Chromium Authors license file <https://src.chromium.org/viewvc/chrome/trunk/src/LICENSE> retrieved in July 2019.)