Referencing within code in software engineering education

Turner, S. and Hill, G

Referencing within Code in Software Engineering Education!

Article · May 2012

2 authors:

Gary Hill
The University of Northampton
20 PUBLICATIONS   **48** CITATIONS

SEE PROFILE

Scott John Turner
The University of Northampton
138 PUBLICATIONS   **271** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   noninvasive glucose monitor View project

Project   leather prediction View project

FINALREFERENCING WITHIN CODE IN SOFTWARE ENGINEERING EDUCATION!

GARY HILL
Department of Computing &
Immersive Technologies, University
of Northampton, Northampton, NN2
6JB, UK

SCOTT TURNER
Department of Computing &
Immersive Technologies, University
of Northampton, Northampton, NN2
6JB, UK

ABSTRACT
Traditionally computer sciences courses will assess software code. It is common and accepted good practice (as in written reports) to reference other sources of appropriate material. However there appears to be no explicit method, recommendation or advice available to computer science tutors and students on a referencing approach! This paper aims to stimulate discussion from peers involved in software engineering education. By discussing the apparent lack of 'referencing within code' advice to students and proposing suggestions for appropriate solutions. This will be based on the authors' experience of assessing code and the current advice given to their students.
KEY WORDS
Software engineering education, referencing, Harvard, computer code.

1 INTRODUCTION

Traditionally computer sciences courses will assess software code at various stages throughout a student's education. It is common and accepted good practice (as in written reports) to reference other sources of appropriate material. With the advent of online electronic submission and the use of plagiarism detection software tools, such as turnitin [1] any similarity of code from the student and external resources are clear highlighted. A similarity index or percentage of the code from other sources in indicated and it is the role of the marking tutor to evaluate the nature of the similarity. As with written reports there is fine line between good and poor academic practice, where a student can use other sources of information to enhance and support their work or pass others work off as their own (by intentionally or unintentionally attributing it clearly). Here the aim is to make an initial proposal of how software engineering lecturers can explicitly advise students on expected 'good practice' for referencing computer code.

1.1 Object Oriented Encourages Re-use.

One of the fundamental principles behind object oriented programming is to re-use code by creating classes, therefore it would be expected that students use class libraries/software development kits (SDKs) available from within a language or external class libraries. Students could/should also be encouraged to write new classes for incorporation/interaction with existing class libraries/SDKs.

1.2 Referencing and Harvard

Within the UK, Harvard appears to be the preferred referencing method recommended to computer science students. The usual 'generic' advice would be that:

> *"A Harvard-style system is the most appropriate system to use for referencing material within your report. Each article should be uniquely identifiable and each reference should be complete so that the reader can trace the article, which is referred to"* [2].

Yet, when code is mentioned, it fails to mention referencing of code:

> *"Documenting software covers a multitude of topics – from commenting program code to writing user guides"* [2].

In other documents, the referencing of code is mentioned, but again normally refers to the citing of software applications/programs and source code within a report, but not within source code:

> *"Software Code*
> *Author (Year) Title of Program (Version Number) [format type] (computer program, software or code, Place of publication: publisher (if available). Available from: URL (if online).*
>
> *E.g.*
> *In Text: (Techsmith, 2008)*
>
> *In Reference list:*
> *TechSmith (2008) Snagit (Version 9.1) [Software] TechSmith Corporation. Available from: http://www.techsmith.com/screen-capture.asp"* [3].

Harvard referencing is widely used and it is proposed that students should use the same referencing system within their code, but slightly adapted to suit coding practices of preceding any code by a header/title block. industry.

2. REFERENCING FOR CODE

Because The proposal from the authors is that referencing for code should follow existing referencing practice, but consider placing the references at the start of the code. The proposed structure of a typical class/source file could be:

- Header
- Disclaimer/copyright
- References
- Code

2.1 Header

A typical header would normally include a few important headings as follows:

```
/**
Program:  Java Graphics Screen Application   <BR>
Filename: GraphicsJFrame.java                <BR>
@author:  © Gary Hill (200WXYZ)              <BR>
Course:   BSc Computing                      <BR>
Module:   Graphics Programming               <BR>
Tutor:    Gary Hill                          <BR>
@version: 1.1                                <BR>
Notes     1.1 Added centreWindow method      <BR>
Date:     28/10/11                           <BR>
*/
```

2.2 Disclaimer/Copyright

A typical disclaimer/copyright section may be considered good practice that confirms that the student is claiming that the code is the work of the student unless otherwise stated. One suggestion is as follows:

```
/*
File: GraphicsJFrame.java

Disclaimer: The following source code is the sole work of the author unless otherwise stated.

Copyright (C) Gary Hill. All Rights Reserved.

*/
```

2.3 References

It is suggested that the references, similar to those at the back of technical reports would follow the header and disclaimer. The referencing within the code itself, using Harvard, could be as follows:

```
/*
References:
The Java SDK is used throughout this application. Other SDKs/Class libraries are as follows:

Java    (2012)    [online]    Java    API    Specification    SE7,    Available    from:
http://docs.oracle.com/javase/7/docs/api/index.html [Accessed 19/01/12] [4]

References used within this application are as follows:

Hill,   G.   J.   (2012)   [online]   Problem   Solving   &   Programming,   Available   from:
http://194.81.104.27/~gary/csy1020/  [Accessed 19/01/12] [5]

Jfreechart   (2012)   [online]   Problem   Solving   &   Programming,   Available   from:
http://www.jfree.org/jfreechart/ [Accessed 19/01/12] [6]
*/
```

In addition to the main referencing block students may need to reference code within the class/source file as follows:
- Class library/SDK
- Method
- Body of the code

2.3.1 Class Library/SDK Referencing

There would need to be a reference to the standard SDK used for the source file, but any used beyond those expected would clearly need to be identified:

```
//<-*****jfreechart (2012) [6] - START
import org.jfree.chart.ChartFactory;
import org.jfree.data.time.TimeSeriesCollection;
import org.jfree.data.xy.XYDataset;
import org.jfree.ui.RefineryUtilities;
```

```
//->***** jfreechart (2012) [6] - END
```

2.3.2   Method Referencing
As methods are self-contained, it is suggested that the reference is given at the start of the method block as follows:

```
//<-***** Hill (2012) [5]
public void centreWindow()
{//Center the window
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = getSize();
…………..
}
```

2.3.3   Referencing from within code
Methods are self-contained, but for referencing of code within a block of code it is suggested that the reference is given at the start and end of the code section for clarity as follows:

```
//<-***** Hill(2012) [5] - START
Transform3D temp = new Transform3D();
viewObjectFromGroup.getTransform(temp);
Transform3D tempDelta = new Transform3D();
tempDelta.setTranslation(new Vector3f(0.0f, 0.0f, -1.0f));
temp.mul(tempDelta);
System.out.println(temp);
float matrix[] = new float[16];        //declare array of 16 floats for matrix
temp.get(matrix);
if (matrix[11] <= 1.0)                 //object front face z = 1
{
        System.out.println("Don't multiply Transform3D at: "+matrix[11]);
}
else                                   //setTransform
{
        viewObjectFromGroup.setTransform(temp);
}
//>-*****Hill(2012) [5] - END
```

3 DISCUSSION
The purpose of this paper was to stimulate discussion from peers involved in software engineering education surrounding the apparent lack of explicit advice to students on the 'referencing of computer code'. It is hoped that the discussions prompted by this paper my lead to the proposal of an appropriate solution/consensus.

Another area for discussion is when students have gained their 'inspiration' from other authors work and have adapted/modified their code or roughly based their code upon this work. The suggestion would be that the work/source should still be cited.

4. CONCLUSIONS
The experience of the authors suggests that a formal code referencing method is needed within Computer Science and has suggested a method, which references code at the start of a class/source file and then references code back to this references block for classes, methods and small sections of code with the class/source file. It is surprising that, when coding and intellectual property rights are fundamental to programming in Computer Science that this issue appears not to have been discussed or raised before, hence the limited sources referenced, but it is hoped this paper will stimulate the production of such references.

REFERENCES
[1] Turnitin. [online] Available from: https://turnitin.com/static/index.php, 2012 [Accessed 19/01/12].
[2] Dawson. C., W. Projects in Computing and Information Systems, A Student's Guide, Second Edition, ISBN-10: 0321263553, Addison Wesley, 2005.
[3] University of Northampton [online] Referencing Guide, The Harvard System, 4th Edition, https://www.northampton.ac.uk/downloads/download/207/referencing-guides/200208, 2012, [Accessed 19/01/12].
[4] Java [online] Java API Specification SE7, Available from: http://docs.oracle.com/javase/7/docs/api/index.html, 2012 [Accessed 19/01/12].
[5] Hill, G. J. [online] Problem Solving & Programming, Available from: http://194.81.104.27/~gary/csy1020/ , 2012 [Accessed 19/01/12].
[6] Jfreechart [online] Problem Solving & Programming, Available from: http://www.jfree.org/jfreechart/, 2012 [Accessed 19/01/12].