A Discussion of Video Capturing to Assist in Distance Learning

Michael Conlon and Vasos Pavlika

University of Westminster, Watford Road, Harrow, Middlesex HA1 3TP, UK

Abstract. This paper discusses video capture as a medium for transferring and reinforcing knowledge using Distance Learning (to be denoted by DL for the remainder of this paper). The area of teaching delivered is computer programming in particular, to the Object Oriented language known as Java, however the techniques introduced are not limited to this sub-discipline of computer science and can be applied to lectures on the theory of databases, formal methods and/or algorithms etc. The software used in this paper is Camtasia which can be applied to the traditional programming languages, including: Java, C++, Visual Basic, C and to the mark-up languages i.e. the Hypertext Markup Language (HTML) and to Javascript. The paper highlights ways of partitioning a teaching demonstration video into different components to achieve multiple views of a particular topic being discussed. This means that students and lecturers are able to view the demonstration repeatedly and more importantly whilst not in a lecture theatre. Once a video has been produced learners are able to follow lecture notes along with the lecturer's discussions at their leisure thus making this method of education a Distance Learning mode, capable of reinforcing learner knowledge. The authors have found that this continual exposure to the lecture material greatly enhances student comprehension, enjoyment and participation. These conclusions were ascertained by conducting experiments in which a comparison of student views (on lectures) were determined i.e. a comparison was made between students taking a class in which Camtasia was used with a class in which Camtasia was not used and the results of the questionnaire/survey are summarised in the conclusions. It was found that the students responded favourably to lectures delivered using the Camtasia environment as the programming ideas could be viewed repeatedly thus reinforcing their knowledge. This was mentioned by the majority of the students (in fact 72% of the students from a sample size of sixty students stated this) and it was felt by the authors that this statistic alone would make the creation and research into further applications of the Camtasia software a suitable, appropriate and worthwhile pursuit. In this paper many programming clips are included with the hope that this illustrates the versatility of Camtasia. The lectures delivered and consequently discussed were presented to a first year undergraduate class in Computer Science studying a variety of Computer Science disciplines including: Artificial Intelligence, Multimedia, Business Computing and e-Commerce. The paper commences with a discussion of two DL environments that the authors are associated with, highlighting points and facilities that are common to both, such as peer-peer discussions, lecturer-student discussions and chat rooms. The paper then goes on to include actual lecture material with associated screenshots

A.A. Ozok and P. Zaphiris (Eds.): Online Communities, LNCS 5621, pp. 432–441, 2009.

using the Camtasia software. The screenshots commence with a demonstration of how to set up the JCreator editor on the University of Westminster server, followed by a demonstration of how the required paths must be set to enable the Java platform to locate all the required classes and libraries to function properly. This is followed by screenshots demonstrating the compilation procedure necessary to successfully run a Java program followed by screenshots on how to debug a typical Java program. In the “Time honoured” fashion the traditional HelloWorld program is also demonstrated and run. This is further complimented by demonstrating the recursive add function using the NetBeans editor. More advanced programming techniques are discussed later in the paper including: the creation of a singleton class with a private constructor and the illustration of the concept of inheritance in Java. Thus the programming techniques introduced are of the OO nature (where the OO denotes Object Oriented) subsequently after these topics have been introduced and discussed feedback from the students is obtained as to the success (or not as the case be) of the effectiveness of using such a method for the delivery of the afore mentioned topics. A discussion of a select few applications of each of these DL environments are also included. The paper reviews the advantages and disadvantages for both students and lecturers alike and the paper also considers many of the difficulties in the recording process that arose. Resource implications are also mentioned relative to the production, i.e. the recording, the delivery and the viewing of the demonstration. The paper concludes with comments from lecturers and students as to the suitability of Camtasia as a teaching method.

1 Introduction

Video capture is a way of capturing the computer screen, all mouse movements and key strokes as well as the voice of the lecturer. This will enable the learner to view the material at a later time and thus become a medium for so-called Distance Learning, a definition of which is given below:

Distance Learning: Wikipedia defines distance education, or distance learning, as a field of education that focuses on delivering education to students who are not physically *on site* to receive their education. Instead, teachers and students may communicate asynchronously (at times of their own choosing) by exchanging printed or electronic media, or through technology that allows them to communicate in real time (synchronously). Distance education courses that require a physical on-site presence for any reason including the taking of examinations is considered to be a hybrid or blended course or program. [9]. Distance Learning (abbreviated to DL for the remainder of this paper) has become a favourable form of education in the 21st century within the Higher Education sector (to be denoted by HE from now on in this paper), as many adult learners in addition to attempting to complete their studies are in full time employment, therefore it is believed that this mode of education merits discussion.

Much literature is available on the use of video capturing in educational environments; an excellent discussion is given by Jensen-Link and Thompson [12]. This particular paper addresses the process for creating digital movies suitable for educational

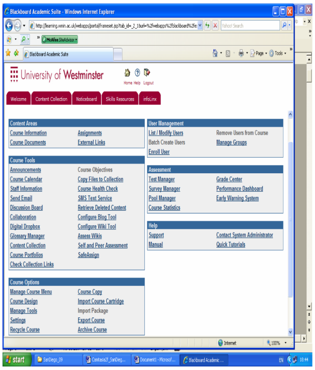
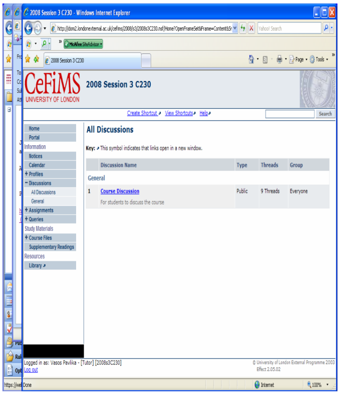


Fig. 1. The Virtual Learning Environment of the University of Westminster



Key: This symbol indicates that links open in a new window.

	Discussion Name	Type	Threads	Group
	General			
1	Course Discussion For students to discuss the course	Public	9 Threads	Everyone

[redacted]

Logged in as: Vasos Pavlika - [Tutor] [2008s3C230]
Log out

© University of London External Programme 2003
Effect 2.05.02

Fig. 2. The Online Study Centre OSC, used at SOAS, University of London

use, and how to manipulate them in order to get the most out of the capturing process. Another excellent discussion appears in the work of Huber [11] where she looks at video capturing used on desktops for use in academia. A wonderful resource to assist first time uses of video capturing in academia can be found on the University of Chicago website [24].

In two of the Institutions that the authors are associated with, Virtual Learning Environments are used in such a way that assignments and learning materials are accessible to students continually, screen shots of these media are shown below in figures 1 and 2 respectively.

Figure 1 represents the virtual learning environment (DL medium) in operation at the University of Westminster (UK) which is known as Blackboard. Blackboard; is a self contained environment that can be used such that all teaching materials and teaching functions (e.g. the setting of assessments, the returning of feedback to students, making group email communications and the making of classroom announcements etc) can be achieved using the platform.

Figure 2 is another Virtual Learning Environment known as the Online Study Centre (OSC) used by SOAS, University of London, all students on the OSC study remotely without attending any lectures. Assignments are submitted via the OSC and marked by tutors of the course. In addition, all learning materials are available on the OSC. Peer to Peer discussion is also possible as is communication between lecturer and student. One may argue that this can also be achieved using traditional email, but with this method all communication is encompassed in a software package and can be referred to at any time.

Hence it clear that DL tools are very popular methods of education, a survey of UK Universities shows that all Universities have analogous DL tools in operation.

2 Video Capture

With Video Capture the recording can be terminated and initiated whilst the capture item is being produced. The item is then stored on some intermediate format which can then be modified, edited, errors can be removed, and additional explanations can be added as well as supplementary descriptions which can be spliced (inserted) in a different order. Post editing, a delivery format must be chosen (e.g. wav, swf, mov, wmv, rm, gif, m4v, Ipod etc) and the particular format file may be placed on a university server to be seen by students in their tutorial sessions or down-loaded to be viewed in their leisure. However the video cannot be easily updated in real time as can a lecture.

This type of software was created and distributed freely by Lotus (1996) however the software was not fully supported by Lotus and more importantly some demonstrations gave a taste and feel of its potential. Later another Software House (Adobe Screen Recorder) showed similar offerings but it was only Camtasia [10] that captured a Java GUI (Panel), Adobe merely showed a black area where the GUI should have been. The above is limited to Personal Computers running the Windows Operating System. There are other capturing software packages for the Linux and the Apple operating systems; however the remainder of this paper will be associated with the Camtasia software.

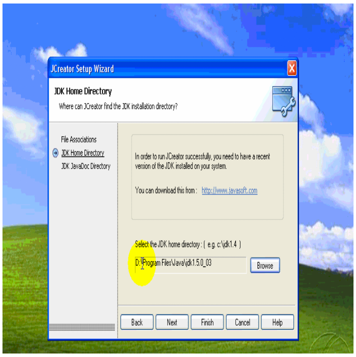
The hardware used was a 3 MHz duel core 4 Giga Byte fast ram computer with a high definition screen such as 1280 by 1024 pixels. Such a screen enables more information to be visible. However the capture area should be smaller than the screen, this enables the lecturer to view information that is not included in the video that is produced.

Thus using Camtasia produces a form of education that one could argue falls precisely in the category of:

Asynchronous Learning: which may be defined as: Any learning event where interaction is delayed over time. This allows learners to participate according to their schedule, and to be geographically separate from the instructor. This could be in the form of a correspondence course or e-learning. Interaction can take use various technologies including a threaded discussion [22].

3 Setting Up the JCreator Editor and the Netbeans Integrated Development Environment (I.D.E.) from a C.D.

Before any DL programming package can be attempted students must be become fully conversant with the software in this case Netbeans [18], JCreator [15] and Java and on how to use the editor to write code leading inevitably to a first program. The authors feel that a systematic and precise approach at the outset gives the students a firm base to build upon. It is daunting to load then set up the software as even the most able students can become confused with the initial setup and the array of screens presented by Netbeans. The authors have prepared many of these starting videos using JCreator and Netbeans for Java. Recently a video showing the installation of C/C++ development tools on a Windows platform [22] using cygwin.com web site [18] was created. Enormous care and precision is required in the production of these videos as the success of programming depends on these initial steps. The authors have found that continuous monitoring of this process leads to a successful and rewarding start to programming modules. Screen shots linking Java SDK [23] to JCreator and Netbeans were developed in 2005 when it was difficult for the students to link the Integrated



JDK Home Directory

Where can JCreator find the JDK installation directory?

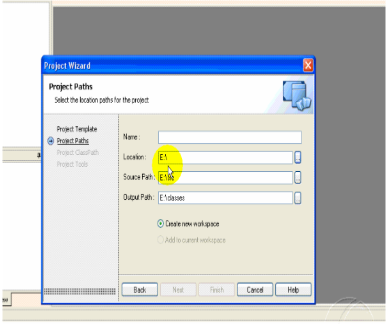
In order to run JCreator successfully, you need to have a recent version of the JDK installed on your system.

You can download this from : http://www.javasoft.com

Select the JDK home directory : (e.g. c:\jdk1.4)

D:\Program Files\Java\jdk1.5.0_03

Fig. 3. Loading JDK with JCreator



Project Paths

Select the location paths for the project

Name :

Location : E:\

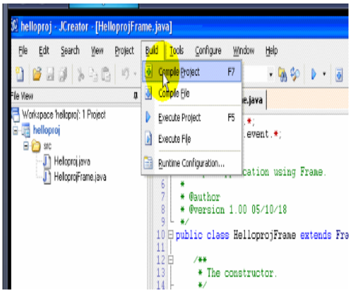
Source Path : E:\src

Output Path : E:\classes

● Create new workspace

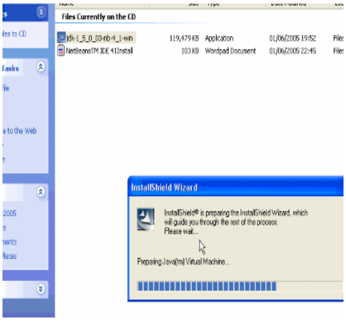
○ Add to current workspace

Fig. 4. Setting the paths to store the Java program for JCreator



```
.*;
.event.*;

cation using Frame.
 *
 * @author
 * @version 1.00 05/10/18
 */
public class HelloprojFrame extends Fra

    /**
     * The constructor.
     */
```

Fig. 5. Compiling a Java program for JCreator



Files Currently on the CD

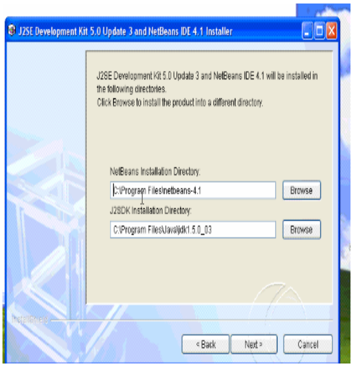
jdk-1_5_0_03-nb-4_1-win	119,479 KB	Application	01/06/2005 19:52	File
NetBeansTM IDE 41Install	103 KB	Wordpad Document	01/06/2005 22:45	File

InstallShield Wizard

InstallShield® is preparing the InstallShield Wizard, which will guide you through the rest of the process. Please wait...

Preparing Java(tm) Virtual Machine...

Fig. 6. Loading Netbeans

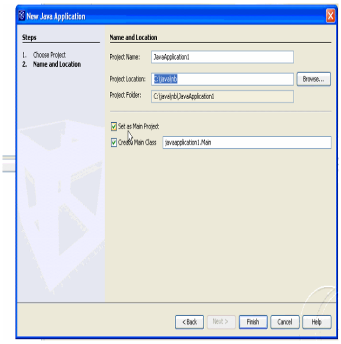

J2SE Development Kit 5.0 Update 3 and NetBeans IDE 4.1 Installer
Browse
Browse
InstallShield
< Back
Next >
Cancel

J2SE Development Kit 5.0 Update 3 and NetBeans IDE 4.1 will be installed in the following directories.
Click Browse to install the product into a different directory.

NetBeans Installation Directory: C:\Program Files\netbeans-4.1

J2SDK Installation Directory: C:\Program Files\Java\jdk1.5.0_03

Fig. 7. Finding the JDK to link with Netbeans



Name and Location

Project Name: JavaApplication1

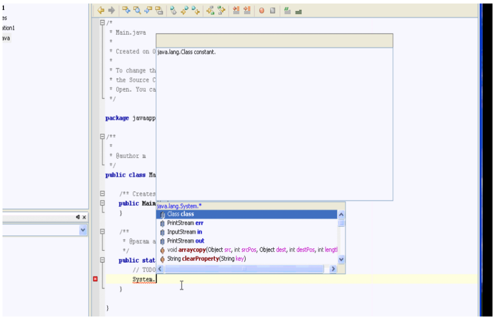
Project Location: C:\java\nb

Project Folder: C:\java\nb\JavaApplication1

☑ Set as Main Project

☑ Create Main Class javaapplication1.Main

Fig. 8. The directory structure for Hello_World with Netbeans



```
/*
 * Main.java
 *
 * Created on 0
 *
 * To change th
 * the Source C
 * Open. You ca
 */

package javaapp

/**
 *
 * @author m
 */
public class Ma

    /** Creates
    public Main
    }

    /**
     * @param a
     */
    public stat
        // TODO
        System.
    }

}
```

java.lang.Class constant.

java.lang.System.*
- Class class
- PrintStream err
- InputStream in
- PrintStream out
- void arraycopy(Object src, int srcPos, Object dest, int destPos, int lengtl
- String clearProperty(String key)

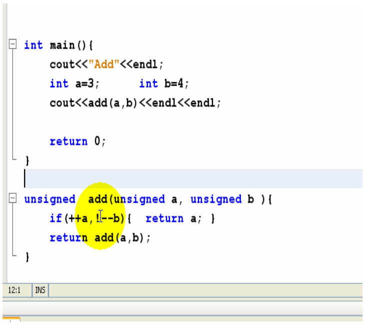
Fig. 9. Netbeans "code complete" for Hello World

Development Environment (I.D.E) to the J.D.K. (The Java Development Kit) compiler. These screen shots only demonstrate the flow of ideas which can be appreciated more fully once the video has been viewed. The following screen shots were made on the University of Westminster network.

4 Classic C++ Swap Demonstration. C++ Recursive Add Function Demonstration

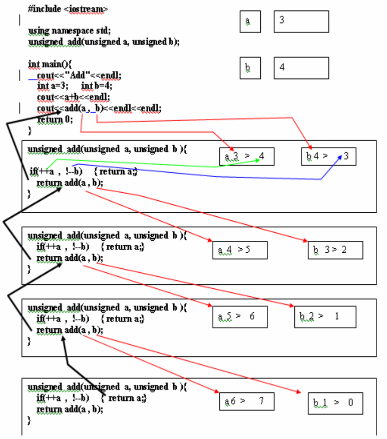
The core of the programming problem must be analysed and the different views of the concept combined in a way to clearly explain the salient points. This process is difficult, however just starting the process, will hone to a reasonable result. For example explaining how a C++ recursive program is connected to the call stack could be covered by a traditional lecturer centred discussion of the code from start to finish as in Eckstein et al [6] and [21]. The authors feel that showing the program running first introduces the terms of the solution before the concept is explained. It may not even be necessary to show the code in the Netbeans editor and proceed to an explanation in an expanded function code diagram.

The next demonstration video is the classic recursive problem using a C++ add() function as discussed in Kernighan [13]. The authors discovered that four basic concepts were required as detailed below. The first shows the code in the Netbeans I.D.E. and is shown running with the result. The second is a Microsoft Word diagram with code and memory allocations. The diagram includes line connections to reaffirm the code with its abstract compiled reality, here it is explained that the function occupies the same place in memory, however the local variables are displaced in time on the call stack. The third and fourth is the code being debugged on a single step basis showing the changing local variables each time the recursive function is called. It must be pointed out that it is a call by value call [21] that enables one variable to be incremented and the other decremented as indicated in the word diagram with code. Later the course demonstrates passing by reference for recursive problems, Stroustrup [21].



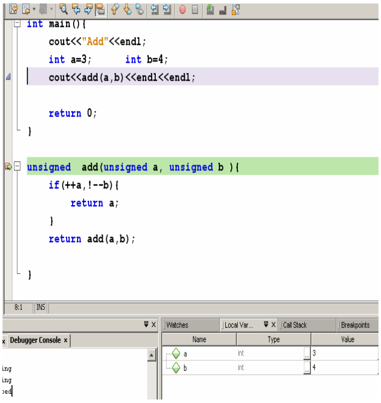
```
int main(){
    cout<<"Add"<<endl;
    int a=3;      int b=4;
    cout<<add(a,b)<<endl<<endl;

    return 0;
}

unsigned  add(unsigned a, unsigned b ){
    if(++a,!--b){  return a; }
    return add(a,b);
}
```

Fig. 10. C++ Netbeans recursive add function



```
#include <iostream>

using namespace std;
unsigned  add(unsigned a, unsigned b);

int main(){
   cout<<"Add"<<endl;
   int a=3;    int b=4;
   cout<<a+b<<endl;
   cout<<add(a, b)<<endl<<endl;
   return 0;
}
unsigned  add(unsigned  a, unsigned  b ){
 if(++a , !--b)    { return a;}
   return add(a , b);
}
unsigned  add(unsigned  a, unsigned  b ){
   if(++a , !--b)    { return a;}
   return add(a , b);
}
unsigned  add(unsigned  a, unsigned  b ){
   if(++a , !--b)    { return a;}
   return add(a , b);
}
unsigned  add(unsigned  a, unsigned  b ){
   if(++a , !--b)    { return a;}
   return add(a , b);
}
```

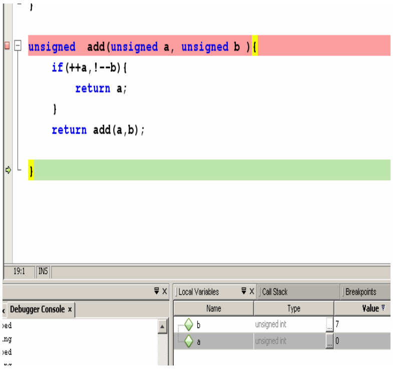
Fig. 11. C++ Netbeans recursive add call diagram



```
int main(){
    cout<<"Add"<<endl;
    int a=3;      int b=4;
    cout<<add(a,b)<<endl<<endl;

    return 0;
}

unsigned  add(unsigned a, unsigned b ){
    if(++a,!--b){
        return a;
    }
    return add(a,b);

}
```

Name	Type	Value
a	int	3
b	int	4

Fig. 12. C++ Netbeans recursive add. Single stepping showing local variables (Starting call a==3, b==4).



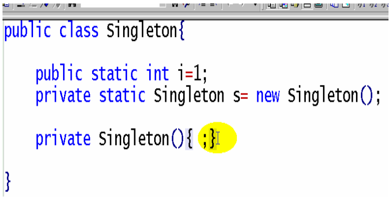
```
}

unsigned  add(unsigned a, unsigned b ){
    if(++a,!--b){
        return a;
    }
    return add(a,b);

}
```

Name	Type	Value
b	unsigned int	7
a	unsigned int	0

Fig. 13. C++ recursive add with debugger (last call a==7 b==0)

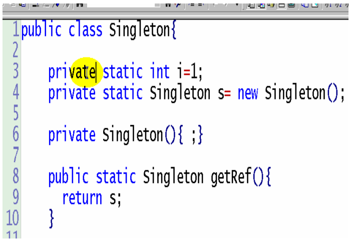
5 Java Video Demonstration of a Singleton Pattern

The video emphasizes the private constructor, Stroustrup[20], Eckel[5] and the public static method that always returns just one unique object. The video then goes on to show how objects that are returned from the singleton class are tested for equivalent addresses. The video addresses the making of the singleton class final and on how the singleton object is instantiated. Lastly on a less technical side, ideas drawn from building houses and towns, Alexander [1] were discussed as a basis to the Standard Template Library Stroustrup [21] and Musser et al [16].



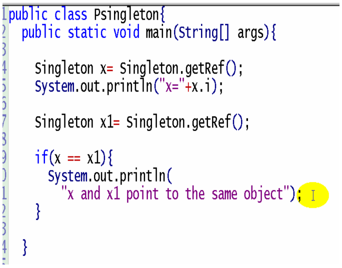
```
public class Singleton{

    public static int i=1;
    private static Singleton s= new Singleton();

    private Singleton(){ ;}

}
```

Fig. 14. Singleton class. With private constructor



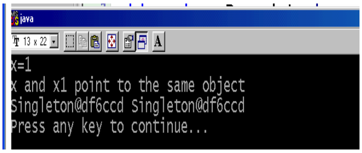
```
public class Singleton{

   private static int i=1;
   private static Singleton s= new Singleton();

   private Singleton(){ ;}

   public static Singleton getRef(){
     return s;
   }
```

Fig. 15. Java singleton class with a static method to return a singleton object



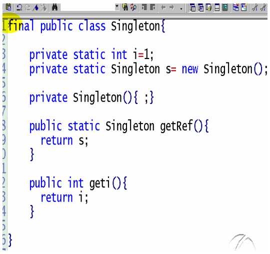
```
public class Psingleton{
  public static void main(String[] args){

    Singleton x= Singleton.getRef();
    System.out.println("x="+x.i);

    Singleton x1= Singleton.getRef();

    if(x == x1){
      System.out.println(
        "x and x1 point to the same object");
    }

  }
```

Fig. 16. Java Singleton driver to compare objects



```
x=1
x and x1 point to the same object
Singleton@df6ccd Singleton@df6ccd
Press any key to continue...
```

Fig. 17. Java Singleton print of object address



```
final public class Singleton{

   private static int i=1;
   private static Singleton s= new Singleton();

   private Singleton(){ ;}

   public static Singleton getRef(){
     return s;
   }

   public int geti(){
     return i;
   }

}
```

Fig. 18. Java final singleton class to prevent inheriting the singleton concept. Hence clone the object.

6 Students' and Lecturers Reaction and Comments

The most able students enjoyed the real time logical mistakes and commented on these when they occurred and offered their solution to the class as the video was paused. These students also felt that they had a something to offer and could follow the video, they also commented that they would watch it on the train as light entertainment or as an alternative to ordinary television. However the weaker students could not enjoy the fun until they had watched the video a number of times with their colleagues who tended to be the more able students. The most encouraging comment came from students who were presented with a problem similar to the video, they mentioned that they would check the video to gain greater insight. The benefit to the authors was that the student's feed back loop did not include the lecturer. The benefit to the students was that the video could be viewed by themselves at their own pace and as already mentioned at their leisure, with or without their colleagues, as many times as they felt was necessary. The feeling was that these demonstrations were an aid to students' successful completion of a software unit; however this is an untested hypothesis.

7 Conclusions

The authors found that running the program and showing the results first gave an anchor to the students. The following salient points were ascertained:

Only the display software (flash, windows media player, Quick time etc) was required on the computer for showing the video. Neither a compiler or an internet connection were required.

Delivery of the concept is key; not the delivery mechanism. Camtasia is only an enzyme (catalyst) of the learning path. Incomplete demonstrations can promote deeper understanding by virtue of the viewer having to join the concepts together. By over analysing a concept the student's thought process can be pre-processed and less flexible, which may hinder future development of the topic Bloom [3].

Video demonstrations can weaken the teacher-student feed back loop and promote independent student thought as compared with real time white board "Chalk and Talk" teaching, Ashcroft [2]. The video demonstrations are fixed in their content and their order and cannot be changed or updated quickly.

Video demonstrations are difficult to compare with Power Point presentations as the latter cannot accommodate the size of the information or the rich multi media and dynamic input provided by video capturing software [7].

It was also found that the setting of programming course work with a video demonstration of the running code helped students visualise what was required. It follows that the students cannot see the code nor gain access to it, Woolhouse [25].

No teaching method fits every topic, teacher, or student, they may be useful and can only be evaluated by time and the educational process.

However it was mentioned by the students that the demonstrations could be viewed again and again which gives rise to abstract reflection of concepts and provides a satisfying medium for understanding the topics being delivered.

These demonstrations have been used on the short courses and found to be useful Knowles [14].

The authors have used this capturing software when conducting Dissertation Voice VIVAS with undergraduate students on computing related degree programmes whilst explaining and altering code. The students were informed, so that no part of their VIVA was undervalued or ignored, Brown et al [4].

The Authors noticed that whilst analysing a concept, then synthesising it to a video demonstration follows Bloom's taxonomy and the skill of the lecturer is to provide the demonstration with only the information required for the target group see Bloom [3].

An example of public sites with video capture is the Integrated Development Editor (IDE): Netbeans. This IDE has been using video demonstrations for the promotion of an open source editor. These videos vary as to the target group and need to be classified, but they do show the way forward in the teaching and the publishing of a package. However they have not tackled the tricky issue of concept presentation as discussed in [8].

Other interesting teaching approaches are discussed in Ashcroft [2] and Rogers [19].

References

1. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language Towns Buildings Construction (1977)
2. Ashcroft, K., Foreman-Peck, L.: Managing Teaching and Learning in Further and Higher Education. Falmer Press, London(1994)
3. Bloom, B.S. (ed.): Taxonomy of Educational Objectives. David McKay Co. (1956)
4. Brown, S., Race, P., Smith, B.: 500 Tips on Assessment (2002)
5. Eckel, B.: Thinking In Java, 2nd edn. Prentice Hall, Englewood Cliffs
6. Eckstein, R., Loy, M., Wood, D.: Java Swing (1998)
7. Education Resources Tecsmith (2008), `http://www.techsmith.com/community/education/default.asp`
8. Flash Demos and video tutorials for NetBeans IDE (2008), `http://www.netbeans.org/kb/55/flash.html`
9. `http://www.usjournal.com/en/students/help/degrees.html`
10. How to write a mouse Listener (Java) (2008), `http://www.java.sun.com/docs/books/tutorial/uiswing/events/mouselistener.html`
11. Huber, M.M.: The use of video capture boards in academia, specifically in desktop presentation and desktop publication. In: Proceedings of the 19th annual ACM SIGUCCS conference on User services (1991) ISBN:0-89791-454-6
12. Jensen-Link, L.A., Thompson, C.: Effective video capture techniques for educational multimedia. In: Proceedings, Frontiers in Education Conference, November 1995, vol. 1, pp. 3a2.30–3a2.33 (1995)
13. Kernighan, B., Ritchie, D.: The C Programming Language (1978)
14. Knowles, M.: The adult learner: a Neglected Species. Gulf Publishing, Houston
15. JCreator I.D.E site (2008), `http://www.jcreator.com/`
16. Musser, D., Derge, G., Saini, A.: STL Tutorial and Reference Guide, 2nd edn. (2006)
17. Netbeans C/C++ installation (September 2008), `http://www.netbeans.org/kb/trails/cnd.html`
18. Netbeans site (2008), `http://www.netbeans.org/`
19. Rogers, A.: Teaching Adults, 2nd edn. Open University Press, Buckingham (1996)
20. Stroustrup, B.: The C++ Programming Language, 2nd edn. Prentice Hall, Englewood Cliffs (1991)
21. Stroustrup, B.: The Design and Evolution of C++. Prentice Hall, Englewood Cliffs (1994)
22. `http://www.delmar.edu/distancelearning/student_success/glossary/glossary-a-c.htm`
23. Sun JDK site (September 2008), `http://java.sun.com/javase/downloads/`
24. `http://drdc.uchicago.edu/what/video-research.html`
25. Woolhouse, M., Jones, T., Rees, M.: Teaching the Post-16 Learner. Northcote House