PROJEK ILMIAH TAHAP AKHIR II

WXES 3182

# Multimedia Training Package on C# Programming Language

Johriah Norfizlina Ismail

WEK 990190

SESI 2001/2002

Supervisor: Dr. P.Sellappan

Moderator: Mr Chiew Thiam Kian

Dissertation submitted by in partial fulfillment of the Requirements for the Bachelors Degree of Computer Science.

# Table Of Contents

## Chapter 2: Literature Review — Page

## Chapter 3: System Analysis

Page

## Chapter 4: System Design

## Chapter 5: System Development and Implementation



## Chapter 6: System Testing

## Chapter 7: System Evaluation and Conclusion

## List Of Tables & Figures

# Abstract

With the introduction of computers, many things have been digitized. The way of learning in the traditional is being replaced by computers, for example Computer Based Training (CBT) and Computer Based Education (CBE).

Multimedia offers exciting and new ways to learn things and also stimulate the users' mind so that they could understand better and make learning more fun.

Computers and multimedia could be the way of learning in time to come and also provide the user with a better and special environment.

For my system, I intended to develop a multimedia training package on C# programming language. C# is a new language developed by Microsoft. It has a very similar syntax of Java and both have the same goal – leading the web-based language.

Competition in Information Technology (IT) field nowadays is very high. That is why I chose to develop multimedia-training package on C#. By doing this, I could learn new technology at early stage. Moreover, I strongly believe that C# will be a commonly use language in future.

This package combines all media features for interactivity. These elements – graphics, images, audios, videos and text, will allow me to develop a good user interface. It is best to consider on color combination, suitable music background and clear voice (explanations).

This multimedia-training package acts similarly with CBT and CBE – that is in CD form. Reason - to allow easy reference and learning tools. That includes easy to carry it anywhere as long as there are an equipped of computers or laptop next to you.

# Acknowledgement

I would like to take this opportunity to express my gratitude towards those who has involved in doing my project.

First of all, I would like to thank my supervisor, Dr. P. Sellappan for his patient on guiding me during my project preparation. Thank you to Pn. Rodziati and Mr. Ibrahim Abu Bakar for their useful advises.

To Mr. Chiew Thiam Kian, thank you for being my moderator and giving me ideas on developing my system.

Finally, thank you to my mom, sisters, brother and my niece for supporting me always. And thank you to my special friend, Abdul Muhaimin bin Abdul Rahim, for giving me such a wonderful ideas and thoughts throughout my project development.

# Chapter 1

# Introduction

# Chapter 1:Introduction.

## 1.0 Project Background.

This project is part of the course requirements in order for me to complete my Bachelor degree in (Software Engineering) and also to contribute to my faculty. The system that I am about to develop is an interactive multimedia-training package on C# programming language. The scope of information that will be included in my training package is limited. This is because, at the time I started with this project, C# and .NET (C# framework) are still in beta version. The purpose to develop this package is to let those who have interest in programming language to be exposed to the latest Internet programming besides the famous beans - Java. This training package will be easier to understand by those who already have programming background on C++, C and Java. The objective of the project is to build a package as one of the teaching device with the use of interactive performance/ presentation by using graphics effects, animations, audios and videos.

## 1.1 Project Definition.

The contents of my multimedia-training package would be C# language. During the development of my system, C# language is still in beta version as well as its' framework/ compiler, .NET Framework. Therefore, the contents of my system are still limited. In my training package, there will be approximately 11 lessons. The lessons of one to eleven will be arranged and structured from basic features to advanced lesson.

Before I proceed with the purpose of my project, let me explain the meaning of multimedia and interactive multimedia in general.

### *1.1.1 Definition Of Multimedia*

Multimedia is defined as the combinations of sound, video, graphics, images and text. Some books defined multimedia as more than that. So here is the extract for the definition of the word multimedia.

According to Damon A. Dean, in his book title "A pocket tour of multimedia on the Internet", he defined multimedia as the technology that combines interactive video, sound, graphics and text. Beyond that, though multimedia is wide open to interpretation.

He also listed in his book that a computer manufacturer might define multimedia in terms of the sound and video hardware capabilities of a computer. In contrast, software companies defined multimedia as the ability to create worlds in which the audience can interact with computers to communicate an idea or to tell a story. Others may focus on the medium that the hardware of content.

But the important point is that multimedia involves two types of experiences that are using and creating. For the type of using experience, a user bought a title and used it for whatever it was designed to do, whether it is a game or educational software. If user creates or edits videos or sounds and combines it into interactive media, this is called creating experience.

Dave Hillman, from Montgomery College explains that multimedia depends on text, pictures, audio and video. Other than that, he also listed out a few key of multimedia concepts

- Multimedia Application Design – the elements and techniques to organize and structure a multimedia application. This also includes storyboarding as a process to design an application.
- Multimedia Authoring Tools – the software programs or tools that are used to create multimedia applications. These range from simple linear presentation programs to complex interactive application builders.
- The Internet – the Internet is rapidly becoming a primary means to deliver multimedia presentation.

- The Multimedia Team – the key players including project manager, designer, graphics artists, programmers, and others who come together to develop a multimedia application.
- The Multimedia Development Process – the steps that are used to plan, define, storyboard, acquire content, author, test, and deliver a multimedia application.

The integration of text, graphics, still and moving images, animation, sound and other medium in a computer controlled environment and the type of information that can be represented, stored, transmitted and processed digitally is multimedia according to P.A Gantt.

### *1.1.2 Interactive Multimedia*

Interactive multimedia is popularly known as hybrid technology. It is a combination for affordable storage and database technology access while it will be manipulate by using authoring multimedia. Interactive multimedia is defined to three criteria:

i. Interactive multimedia can be in any kinds of package that consists of elements combination such as text, graphics, images, animations, video and audio.

ii. These elements will be packaged, integrated and flow together in order for the user to see, browse and analyze these materials through various search and index section.

iii. Interactive multimedia allows user to control reading situation. User will be able to control the screen that they intend to read, able to choose functional buttons that will be given to them, and choose the screen that they want to browse by own the handling-control power in their hand. One of the most important characteristic in interactive multimedia will be its' capacity in using these materials that will produced experience towards users individually.

### *1.1.3 Benefits Of Multimedia*

Benefits of multimedia as outlined in the Jensen and Sandlin's study in 1991 are:

- Multimedia mirrors the way in which the human mind thinks, learns, and remembers by moving easily from words to images to sound, stopping along the way for interpretation, analysis, and in-depth exploration.
- The combination of media elements in a multimedia lesson enables trainees to learn more spontaneously and naturally, using whatever sensory modes they prefer. For example, some people learn best by seeing, others learn best by seeing and hearing, still others learn best through manipulation or kinesthetic (tactile) exercises.
- Combining media elements with well-designed, interactive exercises enables learners to extend their experience to discover on their own, so that they are no longer passive while information is "fed" to them. Additionally, programs may be designed to include immediate feedback and to provide direct reinforcement for correct responses.
- While students may only raise their hands to ask a question so many times, many multimedia programs (expert system) are designed to allow learners pause, branch, or stop for further remediation, exploration, or enhancement opportunities; these interactive qualities encourage non-linear thinking.
- By combining words with pictures, graphics, and audio, multimedia programs enable people with varying levels of literacy and math skill to learn by using sight, hearing, and touch. Evidence suggests that using multimedia segments as context for trainees significantly aids in reading comprehension.
- Instructional technologies help people learn to problem-solve and work in teams, which support the development of interpersonal skill.
- With a multimedia program as assistant, trainers can provide more individualized attention.
- Instructors have time to focus on activities that demand participation while students are able to learn on their own.

Furthermore, humans get their information in the following way:

- More than 80% by sight – of which 20% is remembered
- 11% by hearing – of which 30% is remembered
- 3.5% by smell
- 1.5% by touching and taste.

Out of this, 50% of what is both seen and heard is remembered while 80% of what is seen, heard and done, is remembered.

### *1.1.4 Disadvantages Of Multimedia*

Below is the list of multimedia disadvantages:

- Need high processor speed, memory, disk space and data throughput – high cost
- Those elements like sound, images, or animation and video need higher bandwidth than text files because of the size – the data flow through network will be extremely slow.

## 1.2 Project Purposes

Besides fulfilling the course requirements, by doing this project it gives me self-advantage, whether towards the faculty or the student at self. By doing this project, I am able to learn the latest Internet language (even still in beta version) where I will not be able to learn this in university. Moreover, I strongly believe that this language will definitely be the most commonly used language in future. To be reminding again, this is a Microsoft products we are dealing with.

The reason I chose C# as my title is because it is a very new language and there are not so much training package available on this language. Based on business strategy, we need to grab and apply all new technologies in order to avoid competitions. It is of course also to allow those who are interested to learn this language to be trained without attending any courses or classes. As we know, taking up this kind of courses of private institutions cost a fortune. So, by just having this C# multimedia training package that I am about to build, that consists of basic and advance features, it will allow the user to

make it as a reference any time, anywhere where they need it as long they have an equipped personal computers or laptop with them.

This interactive multimedia-training package can be one of the training and teaching mechanism because it is presented in an interesting, fun and user-friendly environment. Through my research, I realized that this multimedia package should be developed because C# language is fresh and still new for most of us. Malaysian needs to learn new things at an early stage in order to be synchronizing with the fast technology flow. Malaysian as a developing country and with its vision in mind has to compete or at least be parallel with developed countries and in terms of fast growing technologies. Moreover, C# is the first hardcode language that meant for Internet. Do not get confused with this fact, Java is not. History of Java stated that it was known as OWL and meant for logical equipment such as refrigerator and washing machine. Unfortunately, it was an unsuccessful strategy. By the time Sun Microsystems wants to close their language for good, Internet was introduced. So they try to run their language into the net and it was successfully applied. From that day, they change the name of the language to Java.

Back to my multimedia-training package, this package allows users to make it as their reference source without paying for high cost. Besides, buying books in the market are quite expensive especially if it were to be produced by international author. Moreover, the package that I am developing can be kept into CD-ROM so it is easy to bring it anywhere as their learning material as well as their quick reference. Compare to books, most of the time they are thick and so it is so tedious and difficult to bring it anywhere. To put this in the CD is my main objective but I will also put this training package in the Internet where they are allowed to download after pay it online.

Finally, the purpose of developing this package is also to achieve the mission on 'paperless environment' and as we know, by producing books, it requires 100% papers.

## 1.3 Project Objective

As we all know, developing multimedia-training package requires a lot of elements in order to obtain a good and interactive package. Moreover, the information that will be inserted which is C# programming language involves technical skill and good user interface design. So, the objective for the application for the multimedia-training package that I want to develop should be clear on the early stage of its' development. This is vital because this objective will be my reference in developing multimedia training package on C# language.

- Step by step methods on learning the root of C# language until its' advanced features.

  My main objective to develop this system is to teach and train users through the system, about step-by-step methods, the syntax and semantic of C# language and its' application on the net. The lessons will be from basic features to advanced lesson.

- Simple and short user interface and interesting graphics effects.

  If we refer to books, normally the provided manuals are quite complex and too detail. Therefore, I will build this multimedia-training package with simple and understandable presentation. Furthermore, providing the combination of images, audio, animations and video will allow more attractions and for easier understanding of the information that is about to be presented.

- Easy access of information.

  To ease the learning technique through this system, all data access are at user fingertips. The user only needs to click on the mouse to retrieve the desired information. This is clearly showed that it is very different compare to information retrieve from books. By applying this with books, the user has to read and browse through information that is not related to what they are searching for and there is surely a lot of time wasted.

- User-friendly system.

The system that is to be developed is expected to be a user-friendly system plus easy understanding on the presentation, attractive backgrounds and clear fonts. The system will be carefully programmed for easy understanding of instructions by retrieving information quickly. Icons for information retrieval and glossary are provided to assists the user during browsing of this system.

## 1.4 Project Scope.

Scope that will be discussed in this section are divided into two parts that is scope for target user and scope on C# information that will be inserted into this system.

As I define for user expectation for this system would be easy way of learning C# and fast on reaching the advanced features in C#. User's target for the system would be learning methods on C# language easily where this language is going to be one of the most powerful Internet language in future. I believe that most of the time, people always facing difficulties on learning programming language. This is because, their learning tool whether it is a book or a software or even a lecture who gives lecture on programming languages, they do have limitation on fulfilling all users aspects. It is not easy to satisfy everybody who intent to learn languages. For my system, my target users would be non-IT professional. But if they were to be an IT-professional, it is easier to understand this language since they have some basic programming background. It will be more quick learning on C# language if they have a very strong Java knowledge.

The insertion of information into my system will be in order. At early parts of learning, they will be expose to very basic knowledge on C# that includes what is C#, the comparison of C# language with other similar languages, the C# compiler and how to run C#. This will include some information on .NET framework. This framework is the core compiler of C#. After this basic introduction on C#, I will proceed by teaching users the basic coding that includes simple Welcome programming. Next, the syntax and semantic that includes getting familiar with the variables, methods, functions etc. In short, the system will teach users the coding step-by-step. There will be around 11 lessons so far on

teaching C# coding. After that finale lesson, there will be some C# codes example for example creating buttons, background effect, text effect, image and graphics effect, creating forms etc.

## 1.5 Importance Of Project.

Nowadays, technology has been changing from time to time very quickly. Each country will strive hard to leapfrog into Information Age. That of course, Malaysia as a developing country is not excluded. In order to achieve vision of Malaysia – Vision 2020, the community needs to cope with the knowledge of this fast flow Information Technology.

As for my project, I will build a multimedia-training package on new Microsoft Internet language – C#. By doing this, I am able to be part of those who has already involved in this new technology. And I will share with all my friends and colleague in order for them to be in the same boat. It is very important for us to be as the same standard as other technology countries – especially United States. This is because, when we are dealing with computer stream, we are competing worldwide.

Based on this fact, we also need to consider on how to learn new technology. It is very important to have a good reference and learning sources in order to learn and understand quickly: especially if it involves computer language. That is why I consider interactive and easy as my main factors in developing my package.

My expectation on my multimedia-training package I am about to build will be synchronize with user expectation. That is to allow easy and interactive reference. So that user able to understand, learn and adapt to this C# faster and easier.

## 1.6 Software And Hardware Requirements.

### *1.6.1 Hardware*

In order for me to develop multimedia system, hardware that I prefer is:

- Central Processing Unit (CPU)
- VGA Color Monitor
- Multimedia keyboard
- Mouse
- Multimedia speaker
- Printer
- Scanner
- Processor (Intel Pentium 4 consists of 1.3 GB processor)
- 128MB Random Access Memory (RAM)
- Windows 2000 and/or 98 and/or Millennium
- Digital Camera
- Digital Handy Camera

### *1.6.2 Software*

Software that will be used in the system development is:

- Macromedia Director Shockwave Studio version 8
- Adobe Photoshop 6.0
- Flash 5 (may be used/ plan to apply)
- Sound Recorder
- . NET Framework

## 1.7 Project Scheduling

There are few activities level that I did during developing process and finishing this project. Activities are divided into few phases so that I can complete my project in given deadline. Below is the table of activities that was done by me (for the first semester) and please refer to next page to view the Gantt chart report:

| Activities | Date | Milestone (days) |
|---|---|---|
| Choosing title | June 18 | 2 |
| Searching Information | June 20 | 15 |
| Research and investigation | June 20 | 10 |
| Questionnaire | June 20 | 3 |
| Collection of Information | June 20 | 15 |
| Introduction | June 25 | 3 |
| Literature review | July 16 | 4 |
| Methodology | July 16 | 9 |
| System Design | July 27 | 2 |
| Completing Documentation | August 9 | 4 |
| Viva Preparation | August 9 | 5 |
| Viva and proposal submission | August 16 | 1 |
| Preparing Scripts | August 17 | 68 |
| Record Voice over Scripts | November 21 | 15 |
| Preparing Content pages | December 12 | 7 |
| Preparing Main page | December 21 | 5 |
| Preparing Welcome page | December 28 | 10 |
| Testing | January 11 | 3 |
| Modifying | January 16 | 5 |
| Preparing handbook | January 23 | 7 |
| Submit | February 1 | 1 |
| Total of Days | | 194 |

Table 1-1: Schedule Of Activities

## 1.8 Definition Of C#

### *1.8.1 What is C#?*

First of all, the pronunciation of C# is C-sharp not C-hash. That is the first step to learn C#.

Imagine you are creating a new computer language, and you want to solve some of the traditional problems for C and C++ languages; memory leaks, difficulty writing multithreaded applications, static linking, illegal pointer references, overly complex multiple-inheritance rules, and so on. C# is the answer to all these constraints.

C# is a new is a new Object Oriented programming language from Microsoft. A lot of people will say "Oh No" one more language. Yes, one more language. The good news is it is not completely new. If the user is a Java programmer, they should already know good amount of C#, but not complete C#.

. NET Framework is a new programming framework from Microsoft. It promises so many good features to programmers such as automatic memory management, rich class library, cross language inheritance.

In June 2000, Microsoft has announced .NET Framework and C# language (.NET Framework was earlier called as NGWS – Next Generation Windows Services). Ever since, people started discussing about C# and the role it is going to play Windows platform in future. Microsoft has had a close look at Java, C++ and VB, before they designed C#. However, I believe it resembles all three in different ways.

Language wise it is superior to Java, C++ and. The reason is simple. Microsoft took the best o them and added some new features to it, and of course the features of .NET Framework will anyway be there.

### *1.8.2 What is needed to learn C# and use C#?*

To learn C#, it is better if users have some background in programming. If they are already a Java programmer with clear idea about interfaces, packages, exception handling, multithreading, etc., they can pick up this language very quickly. However, there are some minor and major differences even in common topics.

If they are an absolute beginner, they still can learn but it is going to be a long journey. Suggestion by professional usually would be learn Java first and then C#. That will given an easy learning curve for C# and also will give two languages instead of one, which is always better in this ever changing industry.

Regarding software, need to have simple compiler to learn the language. You get command line compiler of C# and even VB.NET with .NET Framework Beta 1, which can be downloaded from www.asp.net, or Visual Studio.NET Beta 1, which can be obtained from Microsoft.

. NET Framework works fine on Windows 2000. It is also possible to use it on Windows 98 but it is not as reliable as the one on Windows 2000. In fact, Microsoft even wants .NET on Windows 95 also down the line. So, all that they need to learn the features of C# is, .NET Framework and a simple editor such as Notepad.

## 1.9 Summary Of Chapter 1

System that I am about to develop is a stand-alone system I interactive multimedia package on teaching C#. The goal of this package is to allow the non-computer people to start learning on latest in Internet language and for the computer professional especially the programmer to be exposed into new language.

Main objective of this package is to design an interface that is easy to be used and understand by users where through their learning there will be insertion of multimedia interactive elements such as graphics, image, audio, animation, text and video.

There will be two considerations on my project scope. Firstly, Information scope that will be inserted into this package will concentrate on the core language by knowing the syntax and semantic of the C# root. By the end of the lesson, some codes example on C# will be shown. Finally, the scope of target user for this multimedia-training package is for those who have interest in learning C# programming language. It can either be a non-IT professional or an IT professional. For those who have a programming background, it will be easier and faster to learn C# programming language – especially if the has Java background.

Generally, importance of the package development is to be put up on the contents and they, are learning the root grammar of the C#. The existence of this package allows the user, whether they are non-IT or IT professional, to be introduced to C# interactively. It is also equally important to be synchronous with the technology country in order to successfully achieve vision of Malaysia – Vision 2020. By developing this package, I am able to share with my friends and colleague to be part of those who has already involve in this new technology of programming language.

Software that will be used to develop this multimedia-training package is mainly Macromedia Director Shockwave Studio 8. Besides that, there is little additional software

that will be used in order to successfully implement and developed my system. They are Photoshop 6.0 and Sound Recorder. I also do plan to add up Flash 5 in my package.

# Chapter 2

# Literature Review

# Chapter 2:Literature Review

## 2.0 Introduction To Literature Review

### *2.0.1 What is a literature review?*

A literature review is an evaluative report of information found in the literature related to your selected area of study. The review should describe, summarize, evaluate and clarify this literature. It should give a theoretical base for the research and help you (the author) determine the nature of your research. Works, which are irrelevant, should be discarded and those, which are peripheral, should be looked at critically.

A literature review is more than the search for information, and goes beyond being a descriptive annotated bibliography. All works included in the review must be read, evaluated and analyzed (which you would do for an annotated bibliography), but relationships between the literatures must also be identified and articulated, in relation to your field of research.

> "In writing the literature review, the purpose is to convey to the reader what knowledge and ideas have been established on a topic, and what their strengths and weaknesses are. The literature review must be defined by a guiding concept (e.g. your research objective, the problem or issue you are discussing, or your argumentative thesis). It is not just a descriptive list of the material available, or a set of summaries."

### *2.0.2 Why do a Literature Review?*

A crucial element of all research degrees is the review of relevant literature. So important is this chapter that its omission represents a void or absence of a major element in research [Afolabi, 1992]. According to [Bourner, 1996] there are good reasons for

spending time and effort on a review of the literature before embarking on a research project.

These reasons include:

- To identify gaps in the literature
- To avoid reinventing the wheel (at the very least this will save time and it can stop you from making the same mistakes as others)
- To carry on from where others have already reached (reviewing the field allows you to build on the platform of existing knowledge and ideas)
- To identify other people working in the same fields (a researcher network is a valuable resource)
- To increase your breadth of knowledge of your subject area
- To identify seminal works in your area
- To provide the intellectual context for your own work, enabling you to position your project relative to other work
- To identify opposing views
- To put your work into perspective
- To demonstrate that you can access previous work in an area
- To identify information and ideas that may be relevant to your project
- To identify methods that could be relevant to your project

As far as the literature review process goes, ultimately the goal for students is to complete their review in the allocated time and to ensure they can maintain currency in their field of study for the duration of their research [Bruce, 1990].

## 2.1 Literature Review Of My Project

There are four types of items that I choose to be literature review. They are book, websites, tools and multimedia training package. As for books and websites, I was able to search all information that I need regarding my project. That includes multimedia tools tutorial – Director 8.0, Flash 5, and also C# programming language plus its' .NET framework. But as for multimedia training package, there are no specific training package

in C# programming language. The one that is most related to it is .NET training course digital seminar. As for tools, there is not so much review that I have done. This is because, I only used the commercialize tools. Moreover, these tools are the tools that I used before to do my past projects. They are Adobe Photoshop 6.0, Macromedia Director 8.0, Sound Recorder, Macromedia Flash 5 and .NET Framework.

Below are the lists of my literature review items that has been done:

### *2.1.1 Literature Review On Books*

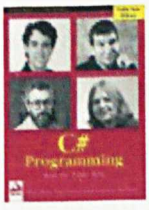
There are only three books of C# that available in Malaysia. These books are revised in order for me to get all the important information on C#. The three books are listed below:


**Programming C# by Jesse Liberty**

*Programming C#* teaches this new language in a way that experienced programmers will appreciate--by grounding its applications firmly in the context of Microsoft's .NET platform and the development of desktop and Internet applications.

Part two of *Programming C#* focuses on development of desktop and Internet applications, including Windows Forms, ADO.NET and ASP.NET. ASP.NET includes Web Forms, for rapid development of web applications, and Web Services for creating objects without user interfaces, to provide services over the Internet.

Part three gets to the heart of the .NET Framework, focusing on attributes and reflection, remoting, threads and synchronization, and streams. Part three also illustrates how to interoperate with COM objects.




**C# Programming with the Public Beta by Burton Harvey, Simon Robinson, Julian Templeman and Karli Watson.**

The goal of this concise volume is to get the reader up to speed on what C# is and how it fits into the Microsoft *vision* for the new .NET. To this end, the book presents a solid tour of .NET features from the Common Language Runtime (and virtual machine) and platform features such as better control of deployment and interoperability with COM, as well as new APIs like ADO.NET (for databases) and ASP.NET (for dynamic Web pages). The other focus of the book is a nicely compact tutorial for C# geared to those with some C++ and/or Java experience. These chapters move quickly through what you'll need to know about C#, from basic data types, flow control, and class design tips, to more advanced features (such as creating and invoking C# objects dynamically or using "unsafe" legacy C++ code from within C#).


**C# Essentials by O'Reilley**

*C# Essentials* is a compact guide. It is an extremely concise--yet entirely thorough--treatment of C#. This is an efficiently packed, language-based guide that's perfect for those with some previous object-oriented programming experience. It covers all C# language keywords, with particular attention to class design constructs. Short code excerpts, rather than full-length programs, are used to illustrate every feature of the language, from basic design types to data types, class design constructs (including all the details of inheritance), and the basics of the .NET Common Language Runtime (CLR). You'll learn how to model classes correctly using the most advanced features (such as reflection and attributes) that help make C# particularly powerful and elegant.

Advanced topics include a guide to making Win32 API calls from within C#, new threading options, and how to interoperate with legacy DLLs and COM objects in C#. At just 200 pages, this text still manages to cover a lot of ground with the specifics of C#, and with many of the features that help give C# its personality as a programming language. It exploiting both basic and advanced features, this title earns high marks as a tutorial for learning Microsoft's latest programming language.

This package explains all elements that can be included into .NET framework. The explanation on C# is only consists of 2 slides and the only things that they talk about on C# is its' functions that support by .NET. C# is developed together with .NET framework. This shows that C# is meant for .NET.

### *2.1.2 Literature Review On Web Sites*

**1. C#@Whiz.com**

**Crisp and Effective** - Being very crisp this tutorial helps you getting started with C# in an extremely short period of time.

**Example based** - Each and every concept is explained with the help of appropriate examples, which help you learn C# in the most efficient way possible.

**Test Engine** - It comes with a test engine and a question bank of 150 questions. Not only you can test your knowledge of C# but also enhance it by going through the exhaustive explanation given for each and every question. It also consists of a ***Diagnostic Test***, which will help you know your strengths and weaknesses, knowing which you can plan your learning accordingly.

**Quick Revision Tips** - Quick revision tips for C#, to make your revision effective and efficient.

**2. csharpindex.com**

It teaches and explains on very basic C# applications. That features include download and install the C# compiler - .Net Framework and install samples. After downloading has been done, it allows user to play with C# coding.

### 3. introduction_to_c#.com

This site explains in detail on all kinds of perspectives on C# applications. It also gives some examples on C# coding depending on the perspective of C# applied. There are 3 different perspectives mentioned in this site:

i. . NET Programming Language
ii. Object-Oriented Programming Language
iii. Intermediate Level Programming Language

This site helps user on differentiate C# programming language at different perspectives.

### 4. msdn.microsoft.com/c#

The site allow user to understand all the features that need to be considerate on C# programming language. It does not contain a technical coding explanations but briefly explain on general features that involves in developing and using C# programming language. The consideration includes:

i. Productivity and Safety:
   - Embraces emerging Web programming standards
   - Eliminates costly programming errors
   - Reduces ongoing development costs with built-in support for versioning

ii. Power, Expressiveness, and Flexibility
   - Better mapping between business process and implementation
   - Extensive interoperability

This site also provide introduction on C# and overview. Finally, a short conclusion at the end of the page is simple and easy enough to understand the reasons of C#.

### *2.1.3 Types Of Multimedia Package Available*

#### I. . NET Training Course – A Digital Seminar on CD-ROM

This package used HTML format that consists of 6+ hours on digital seminar. The main purpose of this training package is to allow user to start developing with Microsoft .NET. This package explains all elements that can be included into .NET framework. The explanation on C# is only consists of 2 slides and the only things that they talk about on C# is its' functions that support by .NET. C# is developed together with .NET framework. This shows that C# is meant for .NET.

Advantages:

- The author, Bertrand Meyer [Appendix A] allow users to trust in buying this package.
- Simple user interface
- Easy in accessing information – good navigation
- Graphics movement is synchronous with the voice/ scripts.
- Provide a handbook that allow user to jot down while listening. It is also provide questionnaires that allow user to determine how far they understand by learning from this package.

Disadvantages:

- Welcome page is not provided
- The introduction page is not interactive
- Did not insert any background music all the way
- Graphics that were used is not attractive and not even related
- The language that were used is not clear (thick German slang) hence hard to understand.

#### II. Microsoft Certified System Engineer (MCSE)

There are four 4 CD-ROMS for MCSE training package. They are:

i. Networking Essentials

ii. NT Server 4.0 for Enterprise

iii. NT Server 4.0

iv. Internetworking with TCP/IP

For this four training package, the concepts that applied for each are consistent. The data that is inserted into the system is very detail and compact. As we know, information is very important for this training package because it is a professional certified program. This training package is not so much of commercial based. It used lotus system for displaying their training courses with the use of Microsoft PowerPoint as user interface and interactions.

Advantages:

- Data and information is compact and detail. This is needed for professional training kit that cost a lot of money.
- Graphics that they applied in their training package are related. These allow users to understand more on information that is presented.
- The movement of graphics and highlighted information are synchronous with the voice explanation.

Disadvantages:

- Language that they used is not professional. The slang of thick Chinese makes users hard to understand his explanations.
- There is not so much navigation that available where users need to load each training course from windows explorer every time they want to change lessons.
- There is no background music all the way.
- They do not provide any welcome screen to attract users in continuing browse the package.

### III. Disney Winnie the Pooh multimedia package

This package is a multimedia package for children age 4-6. Basically this package teaches children how to count, identify shapes and spell. Although the information are

not related to my project – C# programming language, I got attracted to its creativity of the package.

Advantages:

- Graphics and animations are very attractive where it is more to movie-like.
- Good color combination on being synchronous with the concept
- Clear voice and good explanation on giving instructions
- Very attractive in a sense of applying graphics, color, movement and voice.
- Good navigations
- Less text are used
- Simple music background – nice to hear

Disadvantages:

- For those who are used to this package, they could not skip the instructions.
- The long introduction is only suitable for those who want to use the package for the first time.

#### *2.1.3.1 Conclusion on Multimedia Educational Package Available*

| Features/ training package | . NET training course | MCSE | Disney |
|---|---|---|---|
| Compact data | Yes | Yes | Yes |
| Color used | Dull | Nice | Very nice |
| Voice | Not clear | Clear | Clear |
| Navigation | Good | None | Good |
| Language | German slang | Chinese slang | Very clear |
| Music | None | None | Yes |
| Welcome screen | None | None | Yes |
| Graphics movement | Average | Average | Good |

Table 2-1: Comparison on reviewed training package

**Conclusion on comparison table:**

I had chose 3 different types of multimedia training package in order for me to identify a better view on all perspective of users. In a sense of interactivity, Disney multimedia package had attracted most of the user. Although the package is meant for children age 4-6, the attraction came from different age and professions. As for the other two multimedia-training package, the strength lies on its information – compact and detail. The interface is very simple and there are not so much graphics applied. Based on .NET training package, the information to be train is very technical. I believe that is why they could not apply any related graphics. As for MCSE training package, they only apply necessary graphics that helps the user understand more while explaining. There are no additional graphics applied as background. The way I look at it, the more technical it is, and the simpler the interface will be. There is not so much screen for attractions or pleasure – concentrate so much on contents. One that I like most on .Net training courses is the handbook that they provide together with the CD-ROM. It allows the user to jot down notes while listening and the handbook provide questions so that user identifies its ability in understanding .NET.

### *2.1.4 Reviewing Tools*

Before I proceed with the comparison on tools, I will discuss the meaning of *Authoring System.*

#### *2.1.4.1 What is Authoring System*

According to the Multimedia FAQ, an Authoring System is a program which has pre-programmed elements for the development of interactive multimedia software titles. Authoring systems vary widely in orientation, capabilities, and learning curve. Furthermore the FAQ says that "Whether you realize it or not, authoring is actually just a speeded-up form of programming; you don't need to know the intricacies of a programming language, or worse, an API, but you do need to understand how programs work".

Advantages of authoring tools include:

- Ability to do fast prototyping
- Ease of expanding the prototype to a full system
- Ease of use
- Built-in multimedia capabilities
- Less need for programming expertise

Disadvantages of authoring tools include:

- Slow execution – sometimes we do have difficulties on handling too many elements. Especially if the processor, ROM & RAM is not at a high speed plus limited disk storage.
- Poor data handling – conflicts frequent happen in any software or system tool. Usually, authoring tool does not allow a built-in database. As a result, the data is mix up and difficult to control.

There are few methodology used by authoring tools such as (obtained from the multimedia FAQ):

1. Scripting Language
2. Iconic/ Flow Control
3. Frame
4. Card/ Scripting
5. Cast/ Score/ Scripting
6. Hierarchical Object
7. Hypermedia Linkage
8. Tagging

The scripting paradigm is the authoring method closest inform to traditional programming. The paradigm is that of a programming language, which specifies (by filename) multimedia elements, sequencing, hotspots, synchronization, etc. A powerful, object-oriented scripting language is usually the centerpiece of such a system; in-program editing of elements (still graphics, video, audio, etc.) tends to be minimal or

non-existent. Scripting languages do vary; check out how much the language is object-based or object-oriented. The scripting paradigm tends to be longer in development time (it takes longer to code an individual interaction), but generally more powerful interactivity is possible. Since most scripting languages are interpreted, instead of compiled, the runtime speed gains over other authoring methods are minimal. The media handling can vary widely; check out the system with the contribution package formats carefully.

### *2.1.4.2 Comparison Of Tools*

| Tools | Pros | Cons |
|---|---|---|
| **Authorware** Macromedia | Excellent content-creation tools and animation plus special effects capabilities; powerful scripting language for customizing finished courses. | Requires strong programming skills; needs third-party package for information tracking |
| **ToolBook II Assistant** Asymetrix Corp. | Easy-to-use templates ease development; provides good animation capabilities. | No cross-platform development capabilities; |
| **Asymetrix Corp** ToolBook II Instructor | Provides powerful tools for custom course creation; good animation capabilities. | Requires strong programming skills; |
| **IconAuthor** Aimtech Corp. | Powerful tools for custom creation; support Windows and UNIX authoring; good animation capabilities. | Requires strong programming skills; lacks built-in messaging; weak information tracking. |
| **Photoshop 6** Adobe | Have full graphics and photographic editing capabilities, including GIF, transparent GIF and JPEG. | Less web-style shipping graphics than word. |

Continued from page 11....

| Tools | Pros | Cons |
|---|---|---|
| **Fireworks** Macromedia | A really neat feature is being able to define an outline with Fireworks vector tools and the paint it using bitmap functions. Your original vector image can still be modified, too. The Export Preview window shows images regardless of file format. | Not capable of actual animation, importing, n the other hand, is pretty much limited to bitmaps and Adobe Freehand/ Illustrator. |
| **Flash 5** Macromedia | Resolution independent, anti-alias and stream in real-time during playback offering the highest quality viewing experience. | Expensive; scripting are too sophisticated – not suitable for the amateur user. |

Table 2-2: Comparison of tools.

## 2.2 Summary Of Chapter 2

A literature review is an evaluative report of information found in the literature related to your selected area of study. The review should describe, summarize, evaluate and clarify this literature. It should give a theoretical base for the research and help you (the author) determine the nature of your research. A crucial element of all research degrees is the review of relevant literature.

There are four types of items that I choose to be literature review. They are book, websites, tools and multimedia training package.

Books include Programming C# by Jesse Liberty, C# Essentials by O'Reilley, C# Programming with the Public Beta by Burton Harvey, Simon Robinson, Julian Templeman and Karli Watson. Websites that helps me on searching relevant information includes csharpindex.com, intro_to_c#.com, c#@whiz.com and Microsoft.msdn.com/c#.

Comparison of tools and multimedia-training package has been done to identify the best tools and elements to be used and applied on my system. The training package includes .NET training course, MCSE training package and Disney children interactive package. Tools are compared by its' pros and cons in a table to get a better view in identifying which tool to be used.

# Chapter 3

# System Analysis

# Chapter 3: System Analysis

## 3.0 System Development Methodology

For the development of the interactive multimedia training package for C# programming language, I applied the Waterfall Model. The main reason I applied this method is because it has many advantages comparing to other models.

Waterfall model describes its' phases clearly in order for me to develop my system easily. Describing clear phases are very important because these will be my guidance in order to succeed in developing my system. Besides that, information stacking could be redirected although it is impossible to occur. It is very vital for me to make sure that all my work and effort are always in the right path.

Other than advantages mentioned, the other thing that makes me choose Waterfall Model is it provides a clear interaction with system user on what to do. All the needs and desire of the system can be determined at an early stage of the system development. The result of my work progress could be shown to user from time to time so that the system-to-be will always follow and fulfill the desires and needs of system user.

Below is a frame of Waterfall Model that is used in my system development: -

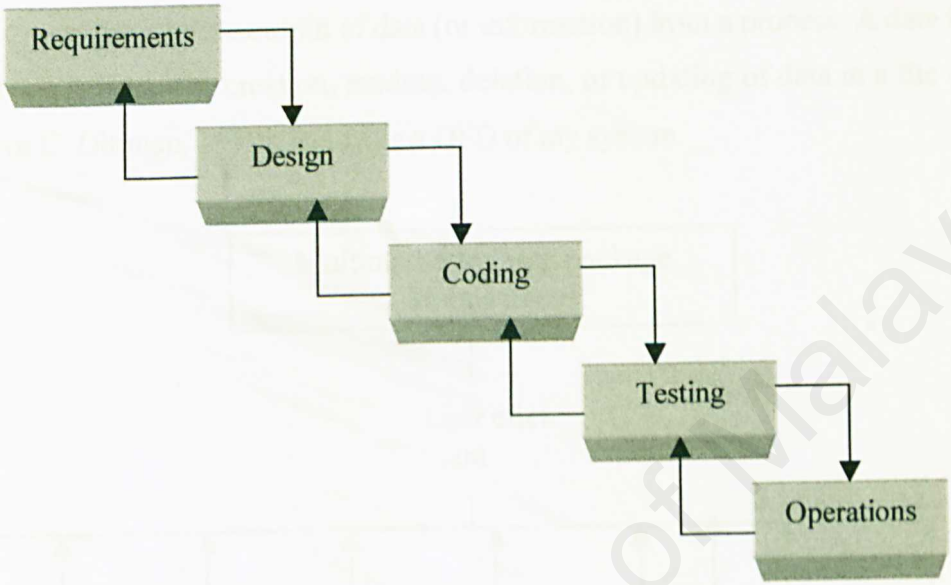
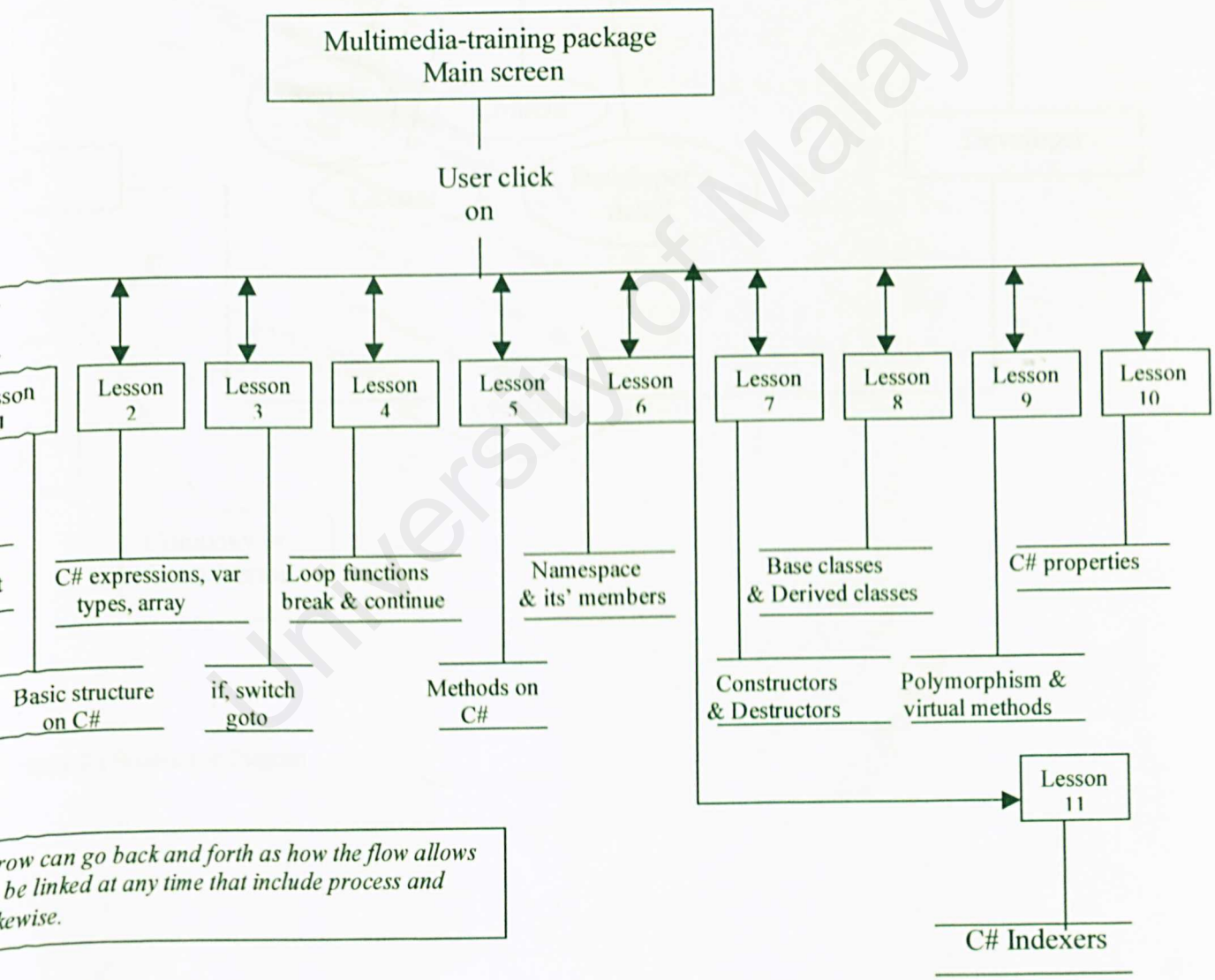



Figure 3.1: Waterfall Model

## 3.1 System's Diagram

### *3.1.1 Data Flow Diagram*

Basically, data flow diagram is a tool that depicts the flow of data through a system and the work and processing performed by that system. Synonyms includes bubble chart, transformation graph and process model. Data flow represents an input of data to a process or the output of data (or information) from a process. A data flow is also used to represent the creation, reading, deletion, or updating of data in a file or database [Kevin C. Dittman, 1999]. Below is a DFD of my system.




*row can go back and forth as how the flow allows be linked at any time that include process and kewise.*

Figure 3.2 Data Flow Diagram

### *3.1.2 Relationship Diagram*

Relationship diagram is drawn to model the system's natural business association that exists between one or more entities. The relationship may represent an event that links the entities or merely a logical affinity that exists between the entities. Below is the system Relationship Diagram that involves two external main factors:

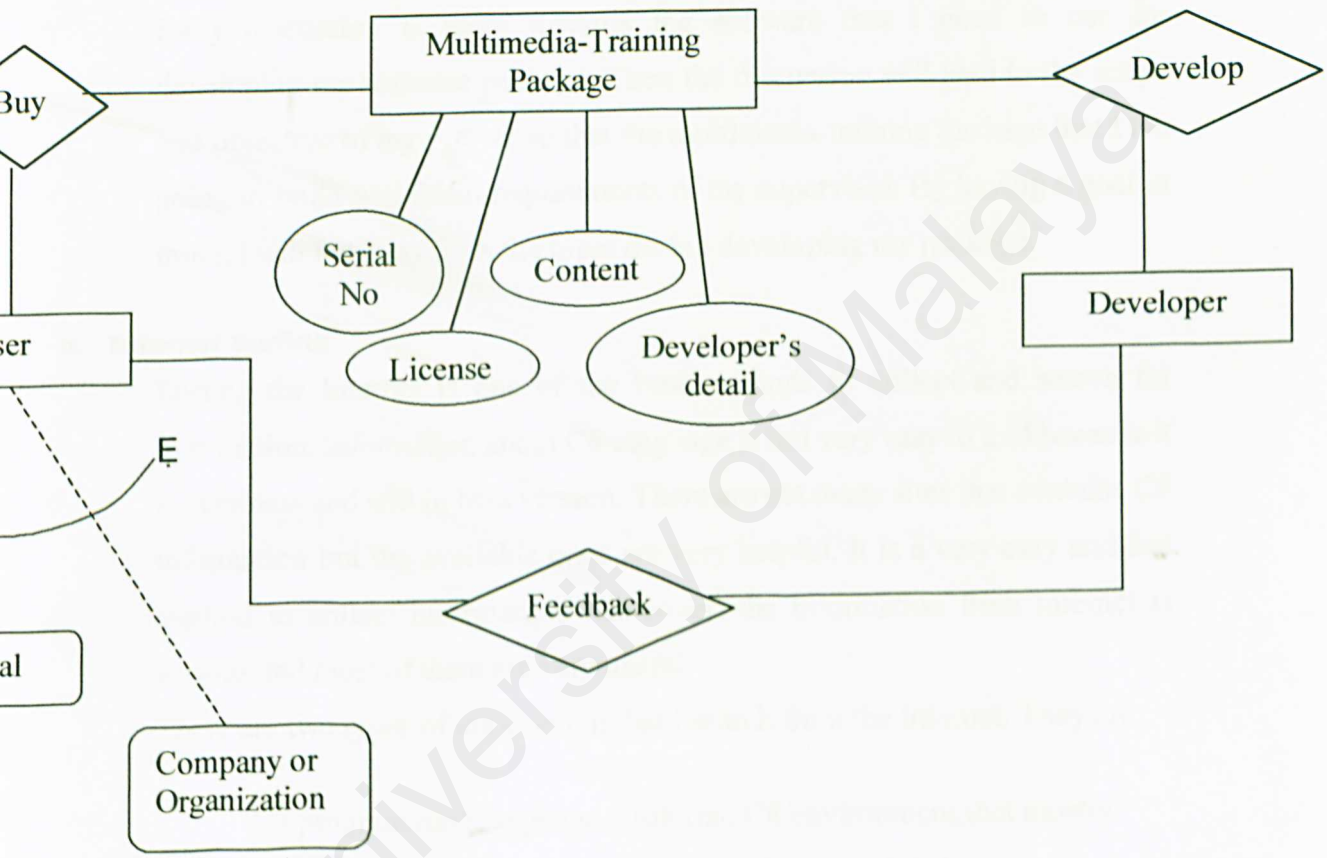



Figure 3.3 Relationship Diagram

## 3.2 Information Collection Technique

There are a few techniques that I used in collecting related information for my system. The related information includes information about software that are going to be use such as Macromedia Shockwave, Flash and also information about C# language. The used technique includes:

**a. Discussion with my supervisor**

Early discussion is more towards the software that I need to use for developing my software package. Then the discussion will lead to the scope and objective of my project so that the multimedia-training package that I am going to build will fulfill requirements of my supervisor. By having a goal in mind, I will not stray from the topic during developing my package.

**b. Internet Surfing.**

Surfing the Internet is one of the best methods to collect and search for information. Information about C# language is not very easy to find because it is very new and still in beta version. There are not many sites that contains C# information but the available ones are very helpful. It is a very easy and fast method to collect information. Moreover, the information from Internet is various and most of them are very useful.

There are two types of information that I search from the Internet. They are:

i. C# programming language – tutorial, C# environment that mostly related to .NET framework and others that are related.

ii. Information on the software that I am going to use – the software that I will be using is new to me so I need to do a lot of learning by example in order for me to come up with ideas.

**c. Reference Books**

Since that C# programming language is new, the reference books available are very limited. It is impossible to obtain C# books from the library. The

available books from bookshops are very few.

Besides that, I also looked reference books on the software that will be used. These will allow me to learn the particular software for my project.

**d. Past Research**

This research is my main reference in developing my system. Although there are different types of multimedia-training package that has already been built, it helps me to get some ideas on how to work on it, how to start and complete my report and how to start in developing my system. The past research is obtained from the documentation room of Faculty of Computer Science and Information Technology. References from different researches allow me to compare and obtain on the information that is needed for my multimedia-training package.

**e. Survey**

In order for me to identify the user's desires and needs, a survey has been done. A survey form was given to 50 respondents from different age group, professions and different academic status that have a computer background or interests. Questions that was asked are mainly about C# programming language and multimedia training packages that they have used. By doing this, it helps me to capture general information on users' preferences in multimedia training package.

**f. Unstructured Interview**

This is more towards general and informal discussion with friends. Usually I will start the discussion by asking general questions on C# with goal in mind. If they were to be off the track, I will redirect them. Without noticing I am actually doing an interview session. By doing this, it allows me to exchange and share ideas with others.

### *3.2.1 Survey*

Survey is an effective way to obtain information on systems. Survey form consists of questions on information that is needed by a system analyst. [Igor Hawryszkiewyez,

1997]. The survey form was titled Multimedia Training package and C# - knowledge, needs and characteristic.

Specifically, this survey was done to:

i. Identify how far Malaysian society from different profession and academic status knows about C# programming language.

ii. Identify elements that should be included into my system to-be following the user needs.

In this survey, most of the questions are in a free-format question where respondents can answer as much as they want based on their opinion in the blank space. *Please refer to Appendix B for sample of survey form.*

#### *3.2.1.1 Sampling*

The survey went on for 8 days. It involves around 50 respondents that were chosen randomly that consist of 3 groups/category. They are students with computer background, professionals in the Information Technology field and others with computer background or interests. The students are from my university, University of Malaya, University of MARA Technology Institute and University of Telecom. While others are my friends working in the computer industry.

#### *3.2.1.2 Survey Output and Analysis*

Results obtained are not conclusive. This is because the free-format questions received mix response. Somehow what were important are the yes and no question to know how far their knowledge on C#. From my survey, I can see that out of 50 respondents, there are only 5 people heard of C# but knows nothing about it. That percentage would be 10%.

The percentage of respondents that has used multimedia-training package before is 12%. From the survey, I was able to identify what are the advantages and disadvantages of the multimedia-training package that was used.

For those who have never used a multimedia-training package before, agree on the format of my system to-be. This is because their perception on learning a new language at early stage are a needed for Malaysian in order for us to be parallel with the fast technology flow around the world especially in competing with high-tech countries such as United States.

The hope from respondents for my multimedia-training package is it should be basic but complete. They also give me some ideas on system user interface to make it interactive and user-friendly. Other important characteristic that was demanded includes easy and simple explanation.

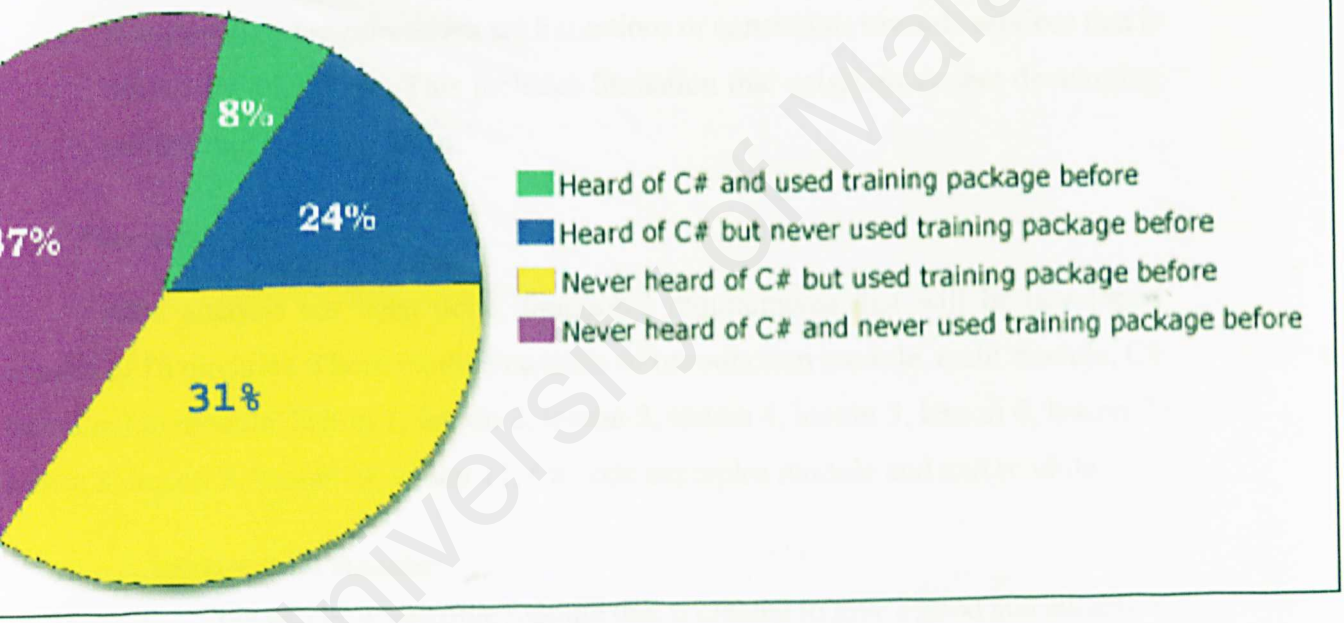



Fig 3.4: Overall conclusion on survey form.

## 3.3 Requirement Analysis

The analysis phase during developing an interactive multimedia-training package is one of the most important phases. In this phase, the entire package initial needs such as identifying the objective, scope, functions, modules and other related information could be defined.

There are two types of requirements. It can either be functional or non-functional [Ian Sommerville, 1996].

- *Functional requirements* are services that are offered by the system, how the system reacts towards the input and the characteristic of the system at different situation.
- *Non-functional requirements* are limitations or constraints towards services that is offered by the system. This includes limitation that exists on system developing process and time limitation.

### *3.3.1 Functional Requirements*

After analysis has been done, functional requirements that will be developed consist of 16 modules. These module consists of introduction module, main module, C# introduction module, lesson 1, lesson 2, lesson 3, lesson 4, lesson 5, lesson 6, lesson 7, lesson 8, lesson 9, lesson 10, lesson 11, C# code examples module and exit module.

i. **Introduction Module**

This will be a welcome module that is created to give a good and attractive start during using my multimedia- training package. This module will briefly explain to the user on what package they are working on and what language they will be learning. I plan to develop my Introduction Module by using Flash 5 that includes attractive graphics, its' movement and songs that will be included to be played synchronously with the displayed graphics.

ii. **Main Module**

After the welcome movie is played, the user will be directed to this module. Its' function is to allow user to make options on what do they want to learn in particular. There will be links to other pages that includes; C# introduction, lesson 1 to lesson 11 and C# code examples. Clicking on the right buttons for the information that are desired will activate the links.

iii. **C# Introduction Module**

This module will explain to users on what is exactly C# programming language. The explanation includes the environment on C#, its' framework that know as .NET framework, why we need to use C#. Basically, it will explain all the basic knowledge on C# to user before they proceed to learn this language.

iv. **Lesson 1:A simple Welcome Program Module**

- Understand the basic structure of a C# program
- Obtain a basic familiarization of what a "Namespace" is
- Obtain a basic understanding of what a class is
- Learn what a "Main" method does
- Learn how to obtain command-line input
- Learn about console input/ output (I/O)

v. **Lesson 2:Expressions, Types and Variables Module**

- Understand what a "Variable" is
- Learn the C# simple types
- Obtain a basic understanding of C# expressions
- Learn what the String type is
- Learn how to use Arrays

vi. **Lesson 3: Control Statements Module**

- Learn the "if" statements
- Learn the "switch" statement
- Learn how "break" is used in "switch" statements
- Understand proper use of the "goto" statement

vii. **Lesson 4: Control Statements-Loop Module**

- Learn the "while" loop
- Learn the "do" loop
- Learn the "for" loop
- Learn the foreach loop
- Complete knowledge of the "break" statement
- Learn how to use "continue" statement

viii. **Lesson 5:Methods Module**

- Understand a structure of a method
- Know the difference between static and instance methods
- Learn to instantiate objects
- Learn how to call methods of an instantiated object
- Understand the 4 types of parameters
- Learn how to use the "this" reference

ix. **Lesson 6: Namespaces Module**

- Understand what Namespace is
- Learn how to implement the "using" directive
- Learn to use "alias" directives
- Understand what are namespace members

x. **Lesson 7:Introduction to Classes Module**

- Implement Constructors
- Know the difference between instance and static members
- Using Destructors
- Familiarization with Class Members

xi. **Lesson 8:Class Inheritance Module**

- Implement Base Classes
- Implement Derived Classes
- Initialize Base Classes from Derived Classes
- Learn how to Call Base Class Members
- Learn how to Hide Base Class Members

xii. **Lesson 9:Polymorphism Module**

- Learn what Polymorphism is
- Implement a Virtual Method
- Override a Virtual Method
- Use Polymorphism in a program

xiii. **Lesson 10:Properties Module**

- Understand what Properties are for
- Implement a Property
- Create a Read-Only Property
- Create a Write-Only Property

xiv. **Lesson 11: Indexers Module**

- Understand what Indexers are for
- Implement an Indexer
- Overload Indexers
- Understand how to implement Multi-Parameter Indexers

**xv. C# Code Examples Module**

Most of the tutorials, either from books or Internet or Multimedia training package do not have this module. They do not give any useful codes that could be applied according to what they learn. This module allows users to try out some codes and apply it on net. By having this module, user will notice that what they have or about to learn from lesson 1 until lesson 11 is worth it. They can see how to apply all their knowledge on C# in order to make the codes useful.

**xvi. Exit Module**

The purpose of this module is to allow user to exit from system.

### *3.3.2 Non-functional Requirements*

After the search for information on modules are done, there are a few non-functional requirements that has to be absorb in this interactive multimedia-training package. The elements that are needed by the user could be obtained from informal and general discussion with friends and results from survey.

**ⓘ User-friendlier Package**

The multimedia-training package that I am going to develop will be very user-friendly. The reason I have to make it extra user-friendly is because; my package consists of a learning programming language. That is not all; even the language that is going to be taught is very new. Moreover, the user target for this package would be for users with and without experience but do have interest on programming language. The important part on this concept would be those without programming language background. First of all, of course the contents should be basic and not confusing. But by doing it user-friendlier, it will not scare away those who have never learn programming language before.

**ⓘ Attractive Interface Design**

The interface design will be one of the elements, which are at the top of my priority list. As we know, human beings always judge by the first impression. Because of this, the interface design needs to be attractive especially for the early screens. There are three criteria that need to be considered on designing the interface; background color, font size and graphics.

**ⓘ Simple and Easy Presentation**

Results from my survey shows that user is hoping for this multimedia-training package to be easy to understand. So, it needs an easy and simple presentation without any dragging and unnecessary explanations. Besides that, each information on the C# programming language will be divided into few parts in order for easy searching and navigation.

**ⓘ Package that need interaction with users**

This multimedia-training package will be developed based on two ways communication between user and computer instructions so that they can surf and search for information that they need quickly by clicking display icons. This package is easy to be navigated; they do not have to have technical computer background. Users only need to click on mouse in order to obtain the information that they desire.

## 3.4 Programming Language

### *3.4.1 Lingo Scripts*

As I mentioned earlier, the software that will be used in developing this interactive multimedia-training package would be Macromedia Director Shockwave Studio 8.0. Director 8 is not a programming language but it is functioning developing tools. There is however a script language that could be added to enhance the system. It is known as Lingo scripts.

Lingo scripts are series of instruction language that is written in Lingo. Lingo scripts can describe some easy action; for example, beep sound when user click on buttons or complex action such as characteristic in moving interactive games on stage. Lingo script is written in Director's script window. The easiest example would be writing programs on beep sound:

```
OnMouseUp me
        beep
end
```

Lingo script are divided into 5 types, they are; primary event scripts, score scripts, cast member scripts, movie scripts and parent scripts.

### *3.4.2 C# Programming Language*

This language is the main language that will be used to teach on my multimedia-training package. I learned the syntax and semantic before I could teach others. The learning process on this language includes the environment of C# - .NET Framework. This .NET Framework became my main compiler in testing C#. Both C# and .NET Framework are Microsoft's products.

## 3.5 System Requirement

Choosing hardware and software is very important in order for the developed system to be successful. Task to choose hardware and software needs to be done particularly to make sure it fulfill system requirement.

### *3.5.1 Software Requirements*

For this multimedia-training package to-be, there are different kinds of software that is suitable to be used. However, I chose Macromedia Director 8.0. There are a few factors that made me choose this software but the main factor is the wide use of this particular software in developing interactive multimedia-training package by the system developer, web designer, students and other individuals. Moreover, I am familiar with

this software. This is because I used to developed software using this tool, only the difference is at that time Macromedia Director is in version 5.5.Macromedia Director 8.0 is main software in producing graphic effects, 3D dimension, video and animation. It also a good and complete solution for Windows 95, Windows 98, Windows NT, Windows 2000, Windows Me, Windows 3.1, Power Macintosh, Macintosh, Internet and Intranet.

However, there is additional software that is added up together with Macromedia Director 8.0. This allows more efficient use on the main software application. The additional software includes Adobe Photoshop 6.0 and Sound Recorder.

#### *3.5.1.1 Macromedia Director 8.0*

Macromedia Director 8.0 is very suitable and the main software that is used to develop interactive multimedia training package and also in developing web sites in the Internet. It allows system developer to create various interactive and fun multimedia presentations. This is because Director 8.0 is consist of various elements or applications that is very important plus its ability to import elements from other software that include Macromedia Flash and Fireworks. The reason is to widen the tools on developing a system.

By using its core elements – 'stage', 'cast', and 'score', Director user will have a full control towards the media elements that includes relationship among animations, sound, video and others. This software eases the system developer in adding basic interactive elements by using authoring tools of 'drag and drop'. To make the system more interactive, Director offered Lingo scripts that can control and program certain movement or other actions. Lingo script is Director language that is quite easy to learn – very English-like.

#### *3.5.1.2 Additional Software*

- **Adobe Photoshop 6.0**

  This software allows me to edit original graphics by adding effects on graphics and so on.

This helps especially on editing scanned graphics. Other editing processes are like change the graphic size, cut certain part of a graphic and edit color of the graphics. It also allows us to draw on its canvas.

- **Sound Recorder**

  This software is used to record sounds and background music. These elements will be added up into my system. The file format of sound recorder is supported by Director 8.0 - *.WAV.

- **Macromedia Flash 5**

  This software will be used to create the Introduction Module. It will be a movie-like introduction. The reason I used this software instead of Director 8.0 itself is because, it is more efficient to make the movie alive by using tools and elements from Flash 5. There are quite a number of tools that provided by Flash does not provided by Director 8.0. Based on the constraints, Flash 5 will be used to create a lively movie-like introduction for welcoming the user.

- **. NET Framework**

  This framework is still in beta version. There are no .NET Framework in CD-ROM available. So, I had to download for hours to get this framework. . NET Framework is a new framework where most of our commonly used language such as VB, ASP, and ADO can be implement in this framework. Here, it is known as VB.NET, ASP.NET and ADO.NET. This framework will treat all the languages that applied equally. It is able to support this languages because of its' wide library. C# is a language that meant for .NET Framework.

This is because C# was developed together with this framework. As I have mentioned, C# enhanced the limitations on C, C++ and VB. So, during the development of .NET Framework, C# is the core language that was used for testing this framework.

### *3.5.2 Hardware Requirements*

There are specifications that need to be considering for Windows platform and Macintosh when using Director 8.0 software for developing interactive multimedia-training package. The characteristic is shown in table below [Phil Gross, 2001]

| Windows | Macintosh |
|---|---|
| Pentium processor or others with same standard, 200Hz or more. | Power PC: Macintosh 68040 (or higher speed) or Macintosh OS 7.1 (or higher speed) |
| 32MB RAM for running synchronous applications. | 32MB RAM for running synchronous applications |
| Direct 3D | Quick Draw 3D |
| 120MB Hard disk for data storage | 85MB Hard disk for data storage |
| 640X480 screen resolution | 640x480 screen resolution |
| 8 bit color (256 color) mode VGA monitor | 8 bit color (256 color) mode SVGA monitor |
| Windows 95, Windows 98, Windows 2000, Windows NT | System 7.5 or latest with Quick Time combination |
| Floppy disk 3.5 high density with 1.44 MB floppy driver | Floppy disk 3.5 high density with 1.44 MB floppy driver |

Table 3-1: Hardware Requirements

## 3.6 Summary of Chapter 3

In developing the interactive multimedia-training package for C# programming language, the system development methodology that was used is the Software Development Life Cycle. This model will describe phases clearly in developing my project.

Information collection technique that is applied consists of few methods. These methods include discussion with my supervisor, Internet surfing, reference from books, past research, survey and unstructured interviews (informal discussion).

Requirement analysis consists of two, they are; functional requirements and non-functional requirements. Functional requirements are consists of 15 modules that need to be in the system. The modules are Introduction module, Main module, C# Introduction module, Lesson 1, Lesson 2, Lesson 3, Lesson 4, Lesson 5, Lesson 6, Lesson 7, Lesson 8, Lesson 9, Lesson 10, Lesson 11 and C# code examples module.

After analysis is done from the collected information, non-functional requirements need to be absorbed in developing this system. It includes user-friendlier system; attractive interface design with the use of multimedia elements and information presented will be easy and simple.

Programming language that will be use is Lingo; script language for Director 8.0 and C# programming language. Other than that, there will be additional software that will be use together with Director 8.0 in order to make more efficient use of the main software application.

# Chapter 4

# System Design

# Chapter 4: System Design

## 4.0 Introduction

In this phase, information and data that was collected for system requirement is used to do logical design for system that I am going to build. Procedures for data input is created so that data that entered into the information system will be correct. Moreover, interactive input will be available for the information system by using form techniques and good screen design.

Parts of logical designing for this application is differentiate the user interface. User interface is vital because it will be a channel that connecting between user and system. This user interface will cover action language and presentation language. These elements will determine the flow of information on how they are sent and accepted between user and system. Combination of these two elements will identify the interface inter-relationship for system application. There are ways on using interface inter-relationship for system application; enter data through keyboard, capture user instructions from menus on screen and click on buttons and icons by using mouse.

This designing phase will also cover file design and database for data storage that is done by application user. A managed database will be the root for system to support information access. Designing for output should fulfill information requirement based on response from users that were collected before.

## 4.1 Program Design

Program designing for development of system to-be, using up-bottom approach design. This type of designing is described as a big system and decomposed to smaller parts [Kendall and Kendall, 1995].

Advantages of up-bottom design are:

i. It avoids development cycle from developing overall system at one time
ii. It avoids developing cycle from misdirect from its' purposes.

### .1.1 Module Design

Modular approach is needed in system designing when up-bottom approach is sed. Modular approach is done by decomposed system into logic modules and nanageable.

Advantages on using modular program are:

i. Easier to write and compile modules because it is able to stand-alone.
ii. Modules are easier to manage for modifying certain functions and not the whole program.
iii. Easier to understand the characteristic of each module. Developer able to take a module and understand its' functions.

n my system, there will be 16 main modules. They are:

- Introduction Module
- Main Module
- C# Introduction Module
- Lesson 1:A simple Welcome Program Module
- Lesson 2:Expressions, Types and Variables Module
- Lesson 3: Control Statements Module
- Lesson 4: Control Statements-Loop Module
- Lesson 5:Methods Module
- Lesson 6: Namespaces Module
- Lesson 7:Introduction to Classes Module
- Lesson 8:Class Inheritance Module
- Lesson 9:Polymorphism Module
- Lesson 10:Properties Module
- Lesson 11: Indexers Module
- C# Code Examples Module
- Exit Module

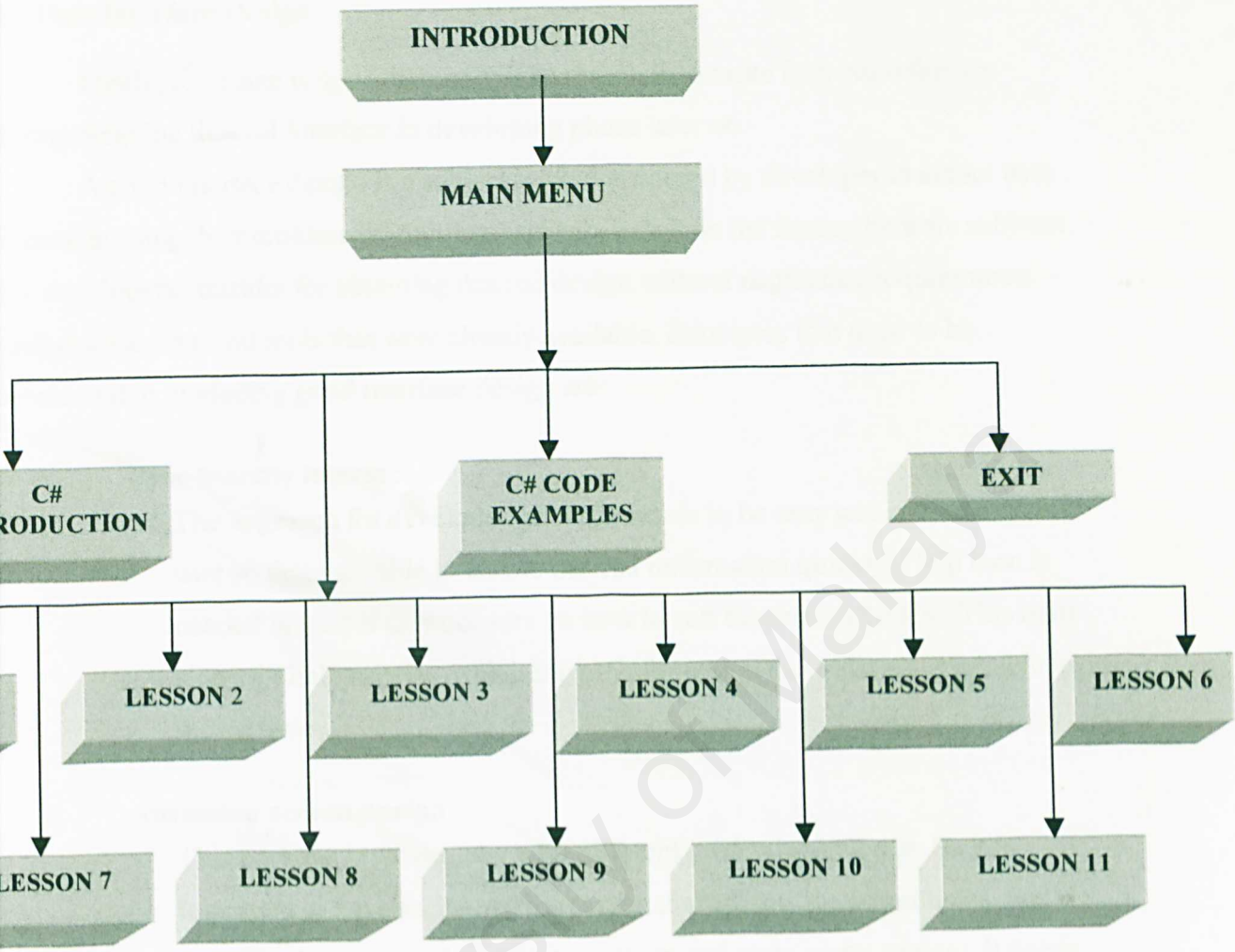



Figure 4.1 System Module Design

## 4.2 User Interface Design

Firstly, this stage is done manually. The reason is to make it as our reference during deigning the real interface in developing phase later on.

A good interface design is a subject to be considered by developer to attract user interest in using their multimedia-training package. Based on the factor, there are subjects that developers consider for obtaining desired design without neglecting requirements such as cost, time and tools that were already available. Principles that need to be considered in producing good interface design are:

i. **User-friendly interface**

The approach for developed package needs to be easy and simple towards user so that user able to access desired information quickly. Help icon is needed in case if user not sure on how to surf the overall system. This help icon will help user by explaining on information that exist and available in the system.

ii. **Attractive screen design**

This package involves user with different profession and age. So it is important to have an interesting screen that attracts the target users. For early display screen (the welcome screen and main menu screen), it needs to be built with high quality graphics effect. As we all know, the first impression is very important to allow them to continue surf the system.

iii. **Screen standards**

Fixing a standard towards each functional button in screen is needed. In developing this package, functional button for screen before will be located at bottom left of the screen while screen after will be located at bottom right of the screen. This standard will be applied on every screen. If there is no standard applied, the unexpected situation will exist and it can cause confusion towards user. Besides that, buttons with same function need to be consistent for each screen for good user interface.

iv. **Recovery**

For each action on functional button on screen need to have services for helping user recover from their mistakes. For example, if user accidentally clicks on exit button, there should be a confirmation message.

v. **Color**

Color also play a main role in attracting user interest especially for reading purposes. For words and sentences that need to be highlighted, it will be brighter and thicker. For background color, a soft color will be used to be synchronous with professional concept. We need to be choosy on applying color in order for displayed images to look perfect and interesting.

vi. **Presentation**

For early screens, high quality of graphics and sound effects will be applied in order to attract user at first sight. The movement of each graphics and words need to be synchronous and suitable with music. When user access to their desired information, most of them concentrate more to interesting text and images that is applied for the particular explanation. Usually, that is what they noticed at first and at most.

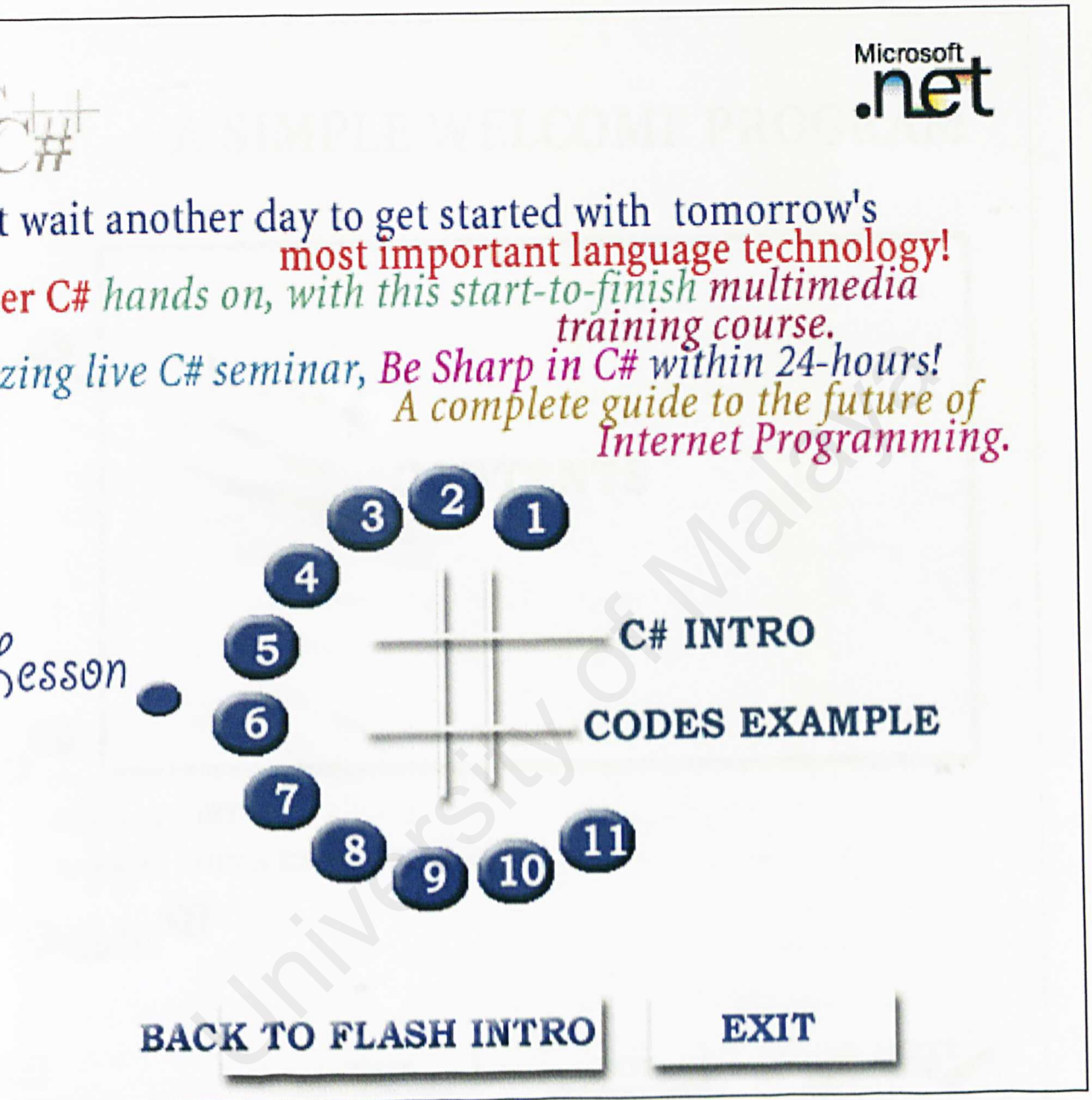
vii. **Help for user**

Interface will gives a convenient towards user for if the user does face difficulties in using the system. That includes difficulty in accessing information. User can refer to help button to overcome these difficulties.

### .2.1 Screen Design

To get used to the package with interactive and multimedia concept, this system eeds to have icons, buttons plus good graphic and sound effects with attractive colors. ince that one of my objective is to develop a good and easy system, it needs to have a lot f icons and buttons to allow convenience towards user in using this system package.

### 2.1.1 *Screen design on* Main Module.




gure 4.2: Screen design on Main Module

lain screen for this system would be designed like above. This screen will appear after lash introduction and they even can go back to flash intro from here. There will be no ack to flash intro' button at other screen. From this screen, user can also access other nformation where the round numbered button represent lessons from lesson 1 until esson 11.

*.2.1.2 Screen design on* **Lesson 1 Module.**


:Screen designs on Lesson 1 Module.

Basically, this will be the design for all screens in the multimedia-training package. Above is the example of screen design on lesson 1. User is allowed to access all the nformation that exists in the system in each screen.

## 4.3 The Expected Outcome Of The System To Be Developed

Here, the discussion is about the expected outcome of my multimedia-training package that I am going to develop. The discussion includes the information source, user interface, concepts and handbook.

- **Information**

  C# has a wide scope of information. I have to make sure that the information to be presented will be easy and compact without leaving any important details behind. The information will be presented in a voice form while presenting C# programs on screen. Basically, all the explanations will be based on C# programs.

- **User Interface**

  Since that types of information is technical – hard to explain and understand; I will not apply too much graphics. This is because, too many graphics will make the user confused. Too many elements on a page will distract user's concentration on important explanations. Graphics that will be applied on each page will be more or less related to the contents.

- **Concepts**

  Based on comparison of CD-ROM that I have made on Chapter 2 – Literature Review; I did mention that Disney training package is the best movie-like that attracts users with different age and professions. But I am not able to do so because the types of information to be presented on my system are totally different with the Disney multimedia package. It does not suitable with too many graphics movement. Moreover, I do not have the skills as the Disney developers have.

  Professional concepts – simple and nice; will be apply on my system package. Not too much color and not too much music. The tone of voice that used for explanations (my voice) will be standardized. I will not speak with difficult slang. Navigations will be my highest priority in allowing user to easily surf my package. Standard buttons will be placed at the same location for each lesson. This is to avoid confusion on searching buttons from a page to another.

**Handbook**

Refining to .NET training course, they provide handbook together with their CD-ROM. I will use this good concept of learning. By having a handbook while using my multimedia-training package, user able to jot down all the important details on spaces provided by the handbook. As told on Chapter 1, human will remember 80% from what is seen, heard and done. This is the main motive for me to do so.

## .4 Summary Of Chapter 4

n this phase, information and data that was collected for system requirement is used to o logical design for system that I am going to build.

rogram designing for development of system to-be, using up-bottom approach design. his type of designing is described as a big system and decomposed to smaller parts.

Modular approach is needed in system designing when up-bottom approach is used. Modular approach is done by decomposed system into logic modules and manageable.

A good interface design is a subject to be considered by developer to attract user interest n using their multimedia-training package. The consideration includes user-friendly nterface, attractive screen design, screen standard, recovery, color, presentation, and help or user.

To get used to the package with interactive and multimedia concept, the systems' screen design needs to have icons, buttons plus good graphic and sound effects with attractive colors. Since that one of my objective is to develop a good and easy system, it needs to have a lot of icons and buttons to allow convenience towards user in using this system package.

Finally, the discussion on the expected outcome would be on the information to be presented, user interface, the concept and handbook that will be provided together with the training package. The discussions are made based on the comparison on available multimedia training package that being made on Chapter 2: Literature Review.

# Chapter 5

# System Development & Implementation

# Chapter 5: System Development and Implementation.

## 5.0 Introduction.

This section will discuss the development strategy as well as type of implementations for the whole package. The coding used in the package will also be discussed and finally the files required in the user system. System implementation is a process that converts the system requirements and design into programming codes. This phase at some point involves some modifications to the previous design.

## 5.1 Development Strategy.

Development environment is of vital importance to any system development. It consists of hardware and software requirements. By using the suitable hardware and software, the process of development able to speed up.

Since this package is about multimedia learning, prototyping should be done. Below are reasons why prototyping is needed:

- Easily change the system should any problem encountered.
- User interface used could well reflect the user needs.
- Able to find any missing functions and requirements.
- Since this package has already existed in the market, prototyping could be used as a template for other implementations.

## 5.2 Coding.

For my multimedia package system, coding or programming language that involves divided to two. They are C# programming language and Lingo scripts. The main factor of this package is the scripts that I will discuss in later section.

### *5.2.1 C# Programming Language*

This language is done basically to teach, as it is the title of my multimedia-training package. There are not so much difficulty in presenting and learn this language although it is still new. This is because the syntax and semantic of the language are

similar to other languages that I have learned. The root of C# language inherits from C and C++ while its' implementation (the way of expressing the codes) are similar to Java.

### *5.2.2 Lingo Scripts*

The scripting engine in Macromedia Director 8.0 – Lingo Scripts, is useful in implementing the various jumping in the package. The jumping here means that the package is divided into smaller files and the communication from one file and another file is through jumping. This is actually calling another file and the presentation will continue from there. This type of jumping was implemented because this will make the file smaller as well as execution time will be a lot faster than putting all the topics in a subject in the same file.

Moreover, these scripts allow the navigation to be interactive as well as the user interface.

## 5.3 C# Coding Contents

### Lesson 1:

**Listing 1-1. A Simple Welcome Program: Welcome.cs**

```
// Namespace Declaration
using System;

// Program start class
class WelcomeCSS
{
   // Main begins program execution.
   public static void Main()
   {
      // Write to console
      Console.WriteLine("Welcome to the C# Interactive Lesson!");
   }
}
```

**Listing 1-2. Getting Command-Line Input: WelcomeYou.cs**

```
// Namespace Declaration
using System;

// Program start class
class WelcomeYou Fizlin
{
```

```
    // Main begins program execution.
    public static void Main(string[] args)
    {
        // Write to console
        Console.WriteLine("Hello, {0}!", args[0]);
        Console.WriteLine("Welcome to the C# Interactive Lesson!");
    }
}
```

**isting 1-3. Getting Interactive Input: InteractiveWelcome.cs**

```
// Namespace Declaration
using System;

// Program start class
class WelcomeYou Fizlin{
    // Main begins program execution.
    public static void Main()
    {
        // Write to console/get input
        Console.Write("What is your name?: ");
        Console.Write("Hello, {0}! ", Console.ReadLine());
        Console.WriteLine("Welcome to the C# Interactive Lesson!");
    }
```

## esson 2:

**isting 2-1. Displaying Boolean Values: Boolean.cs**

```
using System;
class Booleans
{
    public static void Main()
    {
        bool content = true;
        bool noContent = false;

        Console.WriteLine("It is {0} that this lesson provides C# content.", content);
        Console.WriteLine("The statement above is not {0}.", noContent);
    }
}
```

**able 1:**

| ype | Size (in bits) | Range |
|---|---|---|
| byte | 8 | -128 to 127 |
| yte | 8 | 0 to 255 |
| nort | 16 | -32768 to 32767 |
| short | 16 | 0 to 65535 |
| t | 32 | -2147483648 to 2147483647 |

| | | |
|---|---|---|
| int | 32 | 0 to 4294967295 |
| ong | 64 | -9223372036854775808 to 9223372036854775807 |
| ulong | 64 | 0 to 18446744073709551615 |
| char | 16 | 0 to 65535 |

**Table 2:**

| Type | Size (in bits) | Precision | Range |
|---|---|---|---|
| float | 32 | 7 digits | $1.5 \times 10^{-45}$ to $3.4 \times 10^{38}$ |
| double | 64 | 15-16 digits | $5.0 \times 10^{-324}$ to $1.7 \times 10^{308}$ |
| decimal | 128 | 28-29 decimal places | $1.0 \times 10^{-28}$ to $7.9 \times 10^{28}$ |

**Table 3:**

| Category | Operator(s) | Associativity |
|---|---|---|
| Primary | (x) x.y f(x) a[x] x++ x-- new typeof sizeof checked unchecked | left |
| Unary | + - ! ~ ++x --x (T)x | left |
| Multiplicative | * / % | left |
| Additive | + - | left |
| Shift | << >> | left |
| Relational | < > <= >= is | left |
| Equality | == != | right |
| Logical AND | & | left |
| Logical XOR | ^ | left |
| Logical OR | \| | left |
| Conditional AND | && | left |
| Conditional OR | \|\| | left |
| Conditional | ?: | right |
| Assignment | = *= /= %= += -= <<= >>= &= ^= \|= | right |

**sting 2-2. Unary Operators: Unary.cs**

```
using System;

class Unary
{
   public static void Main()
   {
      int unary = 0;
      int preIncrement;
      int preDecrement;
      int postIncrement;
      int postDecrement;
      int positive;
      int negative;
      sbyte bitNot;
      bool logNot;

      preIncrement = ++unary;
      Console.WriteLine("Pre-Increment: {0}", preIncrement);

      preDecrement = --unary;
      Console.WriteLine("Pre-Decrement: {0}", preDecrement);

      postDecrement = unary--;
      Console.WriteLine("Post-Decrement: {0}", postDecrement);

      postIncrement = unary++;
      Console.WriteLine("Post-Increment: {0}", postIncrement);

      Console.WriteLine("Final Value of Unary: {0}", unary);

      positive = +postIncrement;
      Console.WriteLine("Positive: {0}", positive);

      negative = -postIncrement;
      Console.WriteLine("Negative: {0}", negative);

      bitNot = 0;
      bitNot = (sbyte)(~bitNot);
      Console.WriteLine("Bitwise Not: {0}", bitNot);

      logNot = false;
      logNot = !logNot;
      Console.WriteLine("Logical Not: {0}", logNot);
   }
}
```

**ting 2-3. Binary Operators: Binary.cs**

```
using System;
class Binary
{
   public static void Main()
   {
```

```
        int x, y, result;
        float floatResult;

        x = 7;
        y = 5;

        result = x+y;
        Console.WriteLine("x+y: {0}", result);

        result = x-y;
        Console.WriteLine("x-y: {0}", result);

        result = x*y;
        Console.WriteLine("x*y: {0}", result);

        result = x/y;
        Console.WriteLine("x/y: {0}", result);

        floatResult = (float)x/(float)y;
        Console.WriteLine("x/y: {0}", floatResult);

        result = x%y;
        Console.WriteLine("x%y: {0}", result);

        result += x;
        Console.WriteLine("result+=x: {0}", result);
    }
}
```

**sting 2-4. Array Operations: Array.cs**

```
using System;
class Array
{
    public static void Main()
    {
        int[] myInts = { 5, 10, 15 };
        bool[][] myBools = new bool[2][];
        myBools[0] = new bool[2];
        myBools[1] = new bool[1];
        double[,] myDoubles = new double[2, 2];
        string[] myStrings = new string[3];

        Console.WriteLine("myInts[0]: {0}, myInts[1]: {1}, myInts[2]: {2}", myInts[0],
myInts[1], myInts[2]);

        myBools[0][0] = true;
        myBools[0][1] = false;
        myBools[1][0] = true;
        Console.WriteLine("myBools[0][0]: {0}, myBools[1][0]: {1}", myBools[0][0],
myBools[1][0]);

        myDoubles[0, 0] = 3.147;
        myDoubles[0, 1] = 7.157;
        myDoubles[1, 1] = 2.117;
```

```
        myDoubles[1, 0] = 56.00138917;
        Console.WriteLine("myDoubles[0, 0]: {0}, myDoubles[1, 0]: {1}", myDoubles[0, 0],
myDoubles[1, 0]);

        myStrings[0] = "Joe";
        myStrings[1] = "Matt";
        myStrings[2] = "Robert";
        Console.WriteLine("myStrings[0]: {0}, myStrings[1]: {1}, myStrings[2]: {2}",
myStrings[0], myStrings[1], myStrings[2]);

    }
}
```

## esson 3:

**sting 3-1. Forms of the IF statement: IfSelection.cs**

```
using System;
class IfSelect
{
    public static void Main()
    {
        string myInput;
        int myInt;

        Console.Write("Please enter a number: ");
        myInput = Console.ReadLine();
        myInt = Int32.Parse(myInput);

        // Single Decision and Action with brackets
        if (myInt > 0)
        {
            Console.WriteLine("Your number {0} is greater than zero.", myInt);
        }

        // Single Decision and Action without brackets
        if (myInt < 0)
            Console.WriteLine("Your number {0} is less than zero.", myInt);

        // Either/Or Decision
        if (myInt != 0)
        {
            Console.WriteLine("Your number {0} is not equal to zero.", myInt);
        }
        else
        {
            Console.WriteLine("Your number {0} is equal to zero.", myInt);
        }

        // Multiple Case Decision
        if (myInt < 0 || myInt == 0)
        {
            Console.WriteLine("Your number {0} is less than or equal to zero.", myInt);
        }
```

```
        else if (myInt > 0 && myInt <= 10)
        {
           Console.WriteLine("Your number {0} is between 1 and 10.", myInt);
        }
        else if (myInt > 10 && myInt <= 20)
        {
           Console.WriteLine("Your number {0} is between 11 and 20.", myInt);
        }
        else if (myInt > 20 && myInt <= 30)
        {
           Console.WriteLine("Your number {0} is between 21 and 30.", myInt);
        }
        else
        {
           Console.WriteLine("Your number {0} is greater than 30.", myInt);
        }
     }
}
```

**isting 3-2. Switch Statements: SwitchSelection.cs**

```
using System;
class SwitchSelect
{
   public static void Main()
   {
      string myInput;
      int myInt;

      begin:

      Console.Write("Please enter a number between 1 and 3: ");
      myInput = Console.ReadLine();
      myInt = Int32.Parse(myInput);

      // switch with integer type
      switch (myInt)
      {
         case 1:
            Console.WriteLine("Your number is {0}.", myInt);
            break;
         case 2:
            Console.WriteLine("Your number is {0}.", myInt);
            break;
         case 3:
            Console.WriteLine("Your number is {0}.", myInt);
            break;
         default:
            Console.WriteLine("Your number {0} is not between 1 and 3.",
myInt);
            break;
      }
```

```
        decide:

        Console.Write("Type \"continue\" to go on or \"quit\" to stop: ");
        myInput = Console.ReadLine();

        // switch with string type
        switch (myInput)
        {
           case "continue":
              goto begin;
           case "quit":
              Console.WriteLine("Bye.");
              break;
           default:
              Console.WriteLine("Your input {0} is incorrect.", myInput);
              goto decide;
        }
     }
  }
```

## esson 4:

**sting 4-1. The While Loop: WhileLoop.cs**

```
using System;
class WhileLoop
{
   public static void Main()
   {
      int myInt = 0;

      while (myInt < 10)
      {
         Console.Write("{0} ", myInt);
         myInt++;
      }
      Console.WriteLine();
   }
}
```

**ting 4-2. The Do Loop: DoLoop.cs**

```
using System;
class DoLoop
{
   public static void Main()
   {
      string myChoice;

      do
      {
         // Print A Menu
         Console.WriteLine("My Address Book\n");
```

```
            Console.WriteLine("A - Add New Address");
            Console.WriteLine("D - Delete Address");
            Console.WriteLine("M - Modify Address");
            Console.WriteLine("V - View Addresses");
            Console.WriteLine("Q - Quit\n");

            Console.WriteLine("Choice (A,D,M,V,or Q): ");

            // Retrieve the user's choice
            myChoice = Console.ReadLine();

            // Make a decision based on the user's choice
            switch(myChoice)
            {
               case "A":
               case "a":
                  Console.WriteLine("You wish to add an address.");
                  break;
               case "D":
               case "d":
                  Console.WriteLine("You wish to delete an address.");
                  break;
               case "M":
               case "m":
                  Console.WriteLine("You wish to modify an address.");
                  break;
               case "V":
               case "v":
                  Console.WriteLine("You wish to view the address list.");
                  break;
               case "Q":
               case "q":
                  Console.WriteLine("Bye.");
                  break;
               default:
                  Console.WriteLine("{0} is not a valid choice", myChoice);
                  break;
            }

            // Pause to allow the user to see the results
            Console.Write("Press any key to continue...");
            Console.ReadLine();
            Console.WriteLine();
         } while (myChoice != "Q" && myChoice != "q"); // Keep going until the user wants
to quit
      }
}
```

**isting 4-3. The For Loop: ForLoop.cs**

```
using System;
class ForLoop
{
   public static void Main()
   {
      for (int i=0; i < 20; i++)
      {
         if (i == 10)
            break;

         if (i % 2 == 0)
            continue;

         Console.Write("{0} ", i);
      }
      Console.WriteLine();
   }
}
```

**isting 4-4. The ForEach Loop: ForEachLoop.cs**

```
using System;
class ForEachLoop
{
   public static void Main()
   {
      string[] names = {"Fizlin", "Muhaimin", "Mariah", "Irdina"};

      foreach (string person in names)
      {
         Console.WriteLine("{0} ", person);
      }
   }
}
```

## esson 5:

**isting 5-1. One Simple Method: OneMethod.cs**

```
using System;
class OneMethod
{
   public static void Main()
   {
      string myChoice;

      OneMethod om = new OneMethod();

      do
```

```
        {
            myChoice = om.getChoice();

            // Make a decision based on the user's choice
            switch(myChoice)
            {
                case "A":
                case "a":
                    Console.WriteLine("You wish to add an address.");
                    break;
                case "D":
                case "d":
                    Console.WriteLine("You wish to delete an address.");
                    break;
                case "M":
                case "m":
                    Console.WriteLine("You wish to modify an address.");
                    break;
                case "V":
                case "v":
                    Console.WriteLine("You wish to view the address list.");
                    break;
                case "Q":
                case "q":
                    Console.WriteLine("Bye.");
                    break;
                default:
                    Console.WriteLine("{0} is not a valid choice", myChoice);
                    break;
            }

            // Pause to allow the user to see the results
            Console.WriteLine();
            Console.Write("Press any key to continue...");

            Console.ReadLine();
            Console.WriteLine();

        } while (myChoice != "Q" && myChoice != "q"); // Keep going until the user
wants to quit
    }

    string getChoice()
    {
        string myChoice;

        // Print A Menu
        Console.WriteLine("My Address Book\n");

        Console.WriteLine("A - Add New Address");
        Console.WriteLine("D - Delete Address");
        Console.WriteLine("M - Modify Address");
        Console.WriteLine("V - View Addresses");
        Console.WriteLine("Q - Quit\n");
```

```
            Console.Write("Choice (A,D,M,V,or Q): ");

            // Retrieve the user's choice
            myChoice = Console.ReadLine();
            Console.WriteLine();

            return myChoice;
        }
    }
```

## esson 6:

**sting 6-1. The C# Station Namespace: NamespaceCSS.cs**

```
 Namespace Declaration
ing System;

 The C# Namespace
mespace csharp {

// Program start class
class NamespaceCSS {

   // Main begins program execution.
   public static void Main() {

      // Write to console
      Console.WriteLine("This is the new C# Namespace.");

   }
}
```

**ting 6-2. Nested Namespace 1: NestedNamespace1.cs**

```
Namespace Declaration
ng System;

The C# Namespace
nespace csharp{

namespace tutorial {

  // Program start class
  class NamespaceCSS {

     // Main begins program execution.
     public static void Main() {

        // Write to console
        Console.WriteLine("This is the new C# Namespace.");

     }
  }
```

```
}
```

**sting 6-3. Nested Namespace 2: NestedNamespace2.cs**

```
/ Namespace Declaration
sing System;

/ The C# Namespace
amespace csharp.tutorial {

  // Program start class
  class NamespaceCSS {

    // Main begins program execution.
    public static void Main() {

      // Write to console
      Console.WriteLine("This is the new C# Namespace.");

    }
  }
```

**sting 6-4. Calling Namespace Members: NamespaceCall.cs**

```
/ Namespace Declaration
sing System;

amespace csharp {

  // nested namespace
  namespace tutorial {
    class myExample1 {
      public static void myPrint1() {
        Console.WriteLine("First Example of calling another namespace member.");
      }
    }
  }

  // Program start class
  class NamespaceCalling {

    // Main begins program execution.
    public static void Main() {

      // Write to console
      tutorial.myExample1.myPrint1();
      csharp.tutorial.myExample2.myPrint2();

    }
  }
```

```
/ same namespace as nested namespace above
amespace csharp.tutorial {
 class myExample2 {
    public static void myPrint2() {
       Console.WriteLine("Second Example of calling another namespace member.");
    }
 }
```

**sting 6-5. The using Directive: UsingDirective.cs**

```
/ Namespace Declaration
ing System;
ing csharp.tutorial;

/ Program start class
ass UsingDirective {

 // Main begins program execution.
 public static void Main() {

    // Call namespace member
    myExample.myPrint();

 }



 C# Tutorial Namespace
mespace csharp.tutorial {
class myExample {
   public static void myPrint() {
      Console.WriteLine("Example of using a using directive.");
   }
}
```

**sting 6-6. The Alias Directive: AliasDirective.cs**

```
 Namespace Declaration
ing System;
ing csTut = csharp.tutorial.myExample; // alias

 Program start class
ss AliasDirective {

// Main begins program execution.
public static void Main() {

   // Call namespace member
   csTut.myPrint();
   myPrint();

}
```

```
// Potentially ambiguous method.
static void myPrint() {
   Console.WriteLine("Not a member of csharp.tutorial.myExample.");
}


C# Tutorial Namespace
mespace csharp.tutorial {
class myExample {
   public static void myPrint() {
      Console.WriteLine("This is a member of csharp.tutorial.myExample.");
   }
}
```

## sson 7:

**ting 7-1. Example C# Classes: Classes.cs**

```
Namespace Declaration
ng System;

helper class
ss OutputClass {
tring myString;

/ Constructor
ublic OutputClass(string inputString) {
  myString = inputString;


/ Instance Method
ublic void printString() {
  Console.WriteLine("{0}", myString);


/ Destructor
OutputClass() {
  // Some resource cleanup routines



rogram start class
s ExampleClass {

/ Main begins program execution.
ublic static void Main() {

  // Instance of OutputClass
  OutputClass outCl = new OutputClass("This is printed by the output class.");
```

```
        // Call Output class' method
        outCl.printString();

    }
}
```

## Lesson 8:

**Listing 8-1. Inheritance: BaseClass.cs**

```
using System;

public class ParentClass
{
    public ParentClass()
    {
        Console.WriteLine("Parent Constructor.");
    }
    public void print()
    {
        Console.WriteLine("I'm a Parent Class.");
    }
}

public class ChildClass : ParentClass
{
    public ChildClass()
    {
        Console.WriteLine("Child Constructor.");
    }
    public static void Main()
    {
        ChildClass child = new ChildClass();
        child.print();
    }
}
```

**Listing 8-2. Derived Class Communicating with Base Class: BaseTalk.cs**

```
using System;

public class Parent
{
    string parentString;

    public Parent()
    {
        Console.WriteLine("Parent Constructor.");
    }

    public Parent(string myString)
    {
        parentString = myString;
        Console.WriteLine(parentString);
```

```
}

public void print()
{
   Console.WriteLine("I'm a Parent Class.");
}

blic class Child : Parent

public Child() : base("From Derived")
{
   Console.WriteLine("Child Constructor.");
}

public void print()
{
   base.print();
   Console.WriteLine("I'm a Child Class.");
}

public static void Main()
{
   Child child = new Child();
   child.print();
   ((Parent)child).print();
}
```

## esson 9:

**sting 9-1. A Base Class With a Virtual Method: DrawingObject.cs**

```
ing System;

blic class DrawingObject

public virtual void Draw()
{
   Console.WriteLine("I'm just a generic drawing object.");
}
```

**sting 9-2. Derived Classes With Override Methods: Line.cs, Circle.cs, and Square.cs**

```
ing System;

blic class Line : DrawingObject

public override void Draw()
{
   Console.WriteLine("I'm a Line.");
}
```

```
blic class Circle : DrawingObject

 public override void Draw()
 {
    Console.WriteLine("I'm a Circle.");
 }


blic class Square : DrawingObject

 public override void Draw()
 {
    Console.WriteLine("I'm a Square.");
 }
```

**sting 9-3. Program Implementing Polymorphism: DrawDemo.cs**

```
ing System;

blic class DrawDemo

 public static int Main(string[] args)
 {
    DrawingObject[] dObj = new DrawingObject[4];

    dObj[0] = new Line();
    dObj[1] = new Circle();
    dObj[2] = new Square();
    dObj[3] = new DrawingObject();

    foreach (DrawingObject drawObj in dObj)
    {
       drawObj.Draw();
    }

    return 0;
 }
```

## esson 10:

**sting 10-1. An Example of Traditional Class Field Access: Accessors.cs**

```
ing System;

blic class PropertyHolder

 private int someProperty = 0;

 public int getSomeProperty()
 {
    return someProperty;
```

```
}

public void setSomeProperty(int propValue)
{
   someProperty = propValue;
}


blic class PropertyTester

public static int Main(string[] args)
{
   PropertyHolder propHold = new PropertyHolder();

   propHold.setSomeProperty(5);

   Console.WriteLine("Property Value: {0}", propHold.getSomeProperty());

   return 0;
}
```

**sting 10-2. Accessing Class Fields With Properties: Properties.cs**

```
ing System;

blic class PropertyHolder

private int someProperty = 0;

public int SomeProperty
{
   get
   {
      return someProperty;
   }
   set
   {
      someProperty = value;
   }
}


blic class PropertyTester

public static int Main(string[] args)
{
   PropertyHolder propHold = new PropertyHolder();

   propHold.SomeProperty = 5;

   Console.WriteLine("Property Value: {0}", propHold.SomeProperty);
```

```
    return 0;
}
```

**sting 10-3. Read-Only Property: ReadOnlyProperty.cs**

```
ing System;

blic class PropertyHolder

private int someProperty = 0;

public PropertyHolder(int propVal)
{
   someProperty = propVal;
}

public int SomeProperty
{
   get
   {
      return someProperty;
   }
}

blic class PropertyTester

public static int Main(string[] args)
{
   PropertyHolder propHold = new PropertyHolder(5);

   Console.WriteLine("Property Value: {0}", propHold.SomeProperty);

   return 0;
}
```

**ting 10-4. Write-Only Property: WriteOnlyProperty.cs**

```
ng System;

olic class PropertyHolder

rivate int someProperty = 0;

ublic int SomeProperty
{
   set
   {
      someProperty = value;

      Console.WriteLine("someProperty is equal to {0}", someProperty);
   }
```

```
}

blic class PropertyTester

public static int Main(string[] args)
{
   PropertyHolder propHold = new PropertyHolder();

   propHold.SomeProperty = 5;

   return 0;
}
```

## sson 11

**ogram 11-1: An example of an Indexer : IntIndexer.cs**

```
ng System;

ss IntIndexer {
      private string[] myData;

      public IntIndexer(int size) {
               myData = new string[size];

      for (int i=0;i<size;i++) {
               myData[i] = "empty";
              }
      }

olic string this[int pos] {

              get {
                myData[pos];
              }

              set {
                myData[pos[ = value;
              }

              }

tic void Main(string[] args) {
      int size = 10;

      IntIndexer myInd = new IntIndexer(size);

      myInd[9] = "Some Value";
      myInd[3] = "Another Value";
      myInd[5] = "Any Value";
```

```
Console.WriteLine ("\nIndexer Output\n");

for int i=0;i<size;i++)
{
        Console.WriteLine ("myInd[{0}]:{1}",i,myInd[i]);
        }
}
```

**ogram 11-2: Overloaded Indexers : OvrIndexer.cs**

```
ing System;

ss OvrIndexer {
    private string[] myData;
    private int                arrSize;

    public OvrIndexer(int size) {
      arrSize =size;
      myData = new string[size];

      for (int i=0;i<size;i++) {
            myData[1] = "empty";
            }
    }

    public string this[int pos] {

            get {
              return myData[pos];
            }
            set {
              myData[pos] = value;
            }
    }
    public string this[string data] {
    get {
            int count = 0;
    for (int=0;i<arrSize;i++)
    {
            if (myData[i] == data)
            {
              count++;
            }
rn count.ToString();
    }
    set {
            for (int i=0;i<arrSize;I++)
            {
                    if (myData[i] == data)
                    {
                      myData[i] = value;
                    }
```

```
        }
    }

atic void Main(string[] args) {
    int size = 10;
    OvrIndexer myInd = new OvrIndexer(size);

    myInd[9] = "Some Value";
    myInd[3] = "Another Value";
    myInd[5] = "Any Value";

    myInd["empty"] = "no value";

    Console.WriteLine ("\nIndexer Output\n");

    for (int=0;i<size;i++ {

    Console.WriteLine ("myInd[{0}]:{1}",i,myInd[i]);
        }
    Console.WriteLine ("\nNumber of entries:{0}",myInd["no value"]);
        }
    }
```

### *3.1 Structure Programming*

Structure programming is a discipline approach to programming that results in ɔgrams that are easy to read and understand and less likely to contain errors. The ıphasis is on the following accepted program style guidelines to write code that is clear d readable. Obscure tricks and structure programming shortcuts are not taught in this stem package since that it is strongly discouraged. The main advantage of structure ɔgramming is that it is easier to design in the beginning and easier to maintain over the ıg term.

An easy to read codes makes the system package easier to be read and understand. e elements of style include (source code level) documentation, methods for data claration and approach to statement construction. The following are some of the used ling methods:

- Selection of meaningful identifier (variable and labels) names.
- Appropriate comments written in the source code.
- Indentation of codes increases the readability.

## .4 Summary of Chapter 5

evelopment strategy is very important for me to identify hardware and software that ill be used in developing my training package. It allows me to complete my work ithout facing any major difficulties. This includes prototyping to allow user to see the al mock-up model and evaluate. Evaluation will be considered in continuing the evelopment of the real system.

oding that are used in the system package includes C# and Lingo scripts. Since C# is ll in beta version, it takes a longer period to master it and present to user with nfidence. While lingo scripts just take a minor role in my system – most of it was built- script that generated by Director 8 itself.

ructure of programming allows me to present C# codes in more readable and derstandable manner. It eases the ability of understanding by using variables and labels at match with its functions and meaning. Plus, the comments that were inserted roughout the entire lesson for each line of codes allow user to keep track with their aining. Indentation of codes reflects the readability of programming structure.

# Chapter 6

# System Testing

# apter 6: System Testing

## Testing Objectives.

Generally, testing is done to ensure that the training package works properly as pected. This is done by:

Identify, isolate and recover as many bugs as possible. Most programs have bugs, the most insidious of which appear only with unique combinations of data or events.

Demonstrate the functionality of the system appears to be working properly and performance requirements appear to be met.

## 1 Testing Strategies.

ere are 2 division parts of testing:

Module testing or program testing

System testing

### .1 *Module Testing*

r the module testing part, firstly we identify the types of faults.

Algorithmic fault are fault that occurs when a logic or components does not produced the desired output because the processing part contains error. This type of error are easy to spot just by reading through the program or by simulating the types of input the system will receive during its normal working condition. Typical algorithmic faults include:

- Branching too soon
- Branching too late
- Testing for the wrong condition
- Forgetting to initialize variable or set loop invariants
- Forgetting to test for a particular condition

Documentation fault occurs when the documentation does not match with what the program actually does. This will cause other problems in the development of the program later as many modifications done are based on the documentation itself.

Timing or coordination faults occur when the code coordinating these events are inadequate. This fault is hard to detect since it is hard to predict all system state and may be impossible to replicate the fault after it had occurred.

The module testing is divided into unit testing and integration testing. The objective nit and integration testing was to verify that the program was designed as intended by programmer; that is to ensure that the code implemented to design properly.

### *1.1 Unit Testing*

Unit testing means that the various modules or programs are tested individually. unit testing is divided into 2 sections.

The first section is to examine the code and the test the program components. mining the code simply means that the scripting part in the calculation icon is tinized for any errors or problems. For testing the program components, the nching, path or statement testing approaches is done. In branching, the test includes of the branches for every interaction performs. Path, however are quite the same as branch where for any path taken, the test perform will take into consideration of the h taken. Statement testing is just like debugging every statement but in this case, it is ugging every icon in the flow line.

### *1.2 Integration testing*

After the various modules are up to expectation, those modules are merged into final system. Even though the modules are tested, this does not mean that the gration will be a smooth one. After the merging process, the system is put to test to fy the various components could function as a unit. There are a few integration niques that the package could be tested. For example, there are a bottom-up gration, top-down integration big-bang integration, and sandwich integration. Out of e four, this package is put under the big-bang integration. In this integration, all ponents are merged together into a final system and explore to see whether the gration works or not. However, this type of integration is not practical but since the

stem is narrowed, the approach is acceptable. This type of integration makes it difficult find the cause of any failures as well as faults could not be distinguished easily. Even ough this type of approach have disadvantages but it is suitable for this system because e number of components are small.

### *1.2 System Testing*

After all the components are tested individually as well as integrated testing, the erall system testing is done. Here, the primary objective is to ensure that the system es what the customer wants it to perform.

The system is tested on function, performance, acceptance and installation testing. nction testing is based on the functional requirements of the system. Since the number components is small and not too complicated, the testing is done in a very short period.

Next, the performance testing is where functions performance is evaluated. The rformance testing consists of a variety tests. This test depends on the nonfunctional quirements specified.

Configuration tests are conducted on various types of hardware setting and the sults in accordance with the minimum requirements. But the best results will appear nen the hardware configuration is above the minimum requirements. Furthermore, fferent configurations show different results but these are because of the way the fferent PCs video card renders the colors as well as the amount of colors supported by e video card. But this is beyond our control as there is a variety of video cards in the arket.

After the configuration tests, the system is put to timing tests where the respond ne of the system is evaluated. This is because since that it takes time for the animation sound files to load into memory before it could be displayed or played, the respond ne of the system is very important.

Finally, the human factors tests are conducted where the user interface of the stem are examined. User reaction as well as the display screen, the size of fonts used d the ease of use.

As the performance tests finished, the system is put under acceptance testing here user are asked to evaluate the system. The pilot test provides the users with the

stem and let them test the system on their own without following a certain degree of st specs. Since time is limited, the pilot test is experiment by friends and relatives only.

After all the various tests that are conducted, the system is put into final stages of sting which is the installation testing. The testing is done in various PCs in the project o as well as MM1 so that different PCs configuration could be tested.

## 2 Summary of Chapter 6

sting is very important to be done in any tasks. And for this task, testing stage is vital fore the system package is delivering to the customer. Since that there will not be any aintenance services provided, testing need to be done many times to avoid from end-er to detect any errors. However, there will be different versions of the training package om time to time – continuous enhancement will be done.

ere are 2 division parts of testing; they are Module or program testing and System ting. Module testing is divided into another two parts – Unit testing and integration ting, where most of its function is to test individual components of system. After odule completely tested, system-testing takes it turns by test all the functions of system. ey do most of functional requirements testing.

e final testing stage – run system package on various PCs to get precise result in iability of the package to be playable on different platforms.

# Chapter 7

# System Evaluation & Conclusion

# Chapter 7: System Evaluation and Conclusion

## 7.0 Introduction.

The system was evaluated through system testing to identify its strength and limitations; proposal was made for future enhancement. Many problems were encountered during the development of the system and most of them were resolved eventually. Advise from supervisor were definitely a big help on problem solving. Besides that surfing the net, joining the newsgroup, refer to books or even interviewing expert allow me to get the most practical and reasonable solutions.

## 7.1 Problems and Solutions

***Problem 1: Difficulty in defining the system scope at the beginning phase***

As mentioned in Chapter 1, the goal of C# applications are like Java. As for Java, they have different applications that can be applied from it for example, JINI, JNDI, Applets, JAVA API etc. C# similarity, they are consists of Internet based, Windows base, ADO database that can be apply from this particular language. So, it is difficult to identify on what I should insert as contents due to the time limitation.

***Solution:***

Since that C# is still in beta version during my development, I narrowed down the wide scope to basic C# programming language that inherits from C and C++ including Java. Besides that, I will simply give example of codes that helps the user to identify it by themselves and apply it either as Internet based or Windows based application. This will achieve my target audiences to be variety.

***Problem 2: Big file size for packaged and not packaged Director 8 file***

This happened because the size of sound file is big and the drawing tools included with Director 8 is not powerful. So, many images have to be imported into Director 8. This problem bothers even the most seasoned developer [*Consider the advantages of smaller file sizes, Barbara Brown].*

***Solution:***

Since smaller files improved the performance of the whole package, the sound files used are encoded in a lower sample rate and 16-bit format. This makes the sounds appear softer as well as the tone appears longer. Furthermore, the lengths of sound files are for a few seconds only to minimize the file size.

Another solution is to encode the sound into MP3 format. This format makes the usage of more memory since the wave files are compressed, the processor have to do more work. As with MP3 format, certain degree of sound degradation may arise but minimal. Images that are import into Director 8 are converted to GIF or JPEG format. Other than that, the colors of the images are limited to 8-bit.

***Problem 3: No alpha channel support***

Alpha channel are needed to support transparent where the objects blocked by another object are visible. This is different with the transparent in Director 8 that is just for white color. Alpha channel is just another channel of the red, blue and green channel of image. With the addition of alpha channel, the image could be displayed in 32-bit color.

***Solution:***

Develop images or animations that needs alpha channel in Macromedia Flash and embedded the Flash ActiveX control so that Flash files could be shown.

***Problem 4: Lack of font support***

Both the Macromedia Director 8 and Flash have quite a limited type of fonts. For Flash, the fonts are not displayed in a colorful way unlike the Microsoft WordArt. However, the fonts displayed are anti-alias, which makes the fonts appear non-smoothly.

***Solution:***

Develop text in the normal format. Text that was developed in Adobe Photoshop is to imply effects on it – import is needed into either Flash or Director 8. For the anti-alias, the font used should be small so that the font appears smoothly.

***Problem 5: Lack of scripting option***

Even though Director 8 itself has a scripting language – Lingo, it is not powerful enough. It does not offer more advanced programming option like a full-blown programming language. Furthermore, the mouse event listener is not fully implemented in scripting. For example, the mouse over event or mouse out event could not be implemented without using the interaction icons and duplicating the various mouse events for a button.

***Solution:***

The easiest solution is to use many icons to stimulate the various mouse events. For example, using the hotspot twice to capture the mouse over and mouse click event. Another solution is to use button to substitute the hotspot.

***Problem 6: Lack of function to support interactive features***

Since it does not support alpha channel, various objects that need to be displayed transparent will have to do with non-transparent effects. The graphics tools that are offered are not powerful enough and are fairly limited to a few choices only.

***Solution:***

Developed the interactive features in other programs for example in Adobe Photoshop.

***Problem 7: Difficulty in creating good user interface for the learning package***

This is because there are many aspects of good designing skills that need to be met. It has to be professional with the contents that need to be presented. The various aspects includes colors, graphics, animations, navigation and, of course ease of use.

***Solution:***

Fix concepts before gather sources for designing. Get opinions and suggestions from variety users. Prototypes interface and test it to different kind of people – get comments and improve eventually. As far as I am concern, the lifecycle of designing takes a long period to end with one good ides.

***Problem 8: Recording voice***

At first I did my recording on normal Windows voice recorder. Then I realize there are so much distortion and noises that I am not able to fix it. There are no tools in Windows voice recorder that allow me to eliminate noises and distortions.

***Solution:***

By getting advices from friends, I able to eliminate the noise minimally, but the sound are still very much low quality. I searched another voice recording software. I end up with Cool Edit that allow me to eliminate distortions and noises plus obtaining a very good sound quality that even allows me to save it as MP3 format for compressions purposes.

***Problem 9: Inserting Flash***

The first time I inserted Flash movie in Director 8, it does not functioning as well as in Flash player. The graphics, sounds and music coordination do not synchronized as how it supposed to be.

***Solution:***

I had to check each frame and scores of Director 8 and adjust its tempo for the scores to allow synchronization of music, graphics and sound coordination.

## 7.2 Features and Strength

The following illustrates the key strengths of C# Interactive Training Package:

i. **User-friendly**

C# Interactive Training Package has a user-friendly and consistent interface for the ease of users. A set of standard GUI has been implemented and has been implemented. It also has been set to allow the users to browse in a very short time.

ii. **Run in all PC with CD-ROM**

Since that I created a projector, which is .exe file extension, the package is can easily run in any PC as long as it is CD-ROM attached without any specific requirements.

**iii. Fast Response**

The C# Interactive Training Package is in CD form. All the response are depending on the speed of users' CD-ROM. Nowadays CD-ROM technology can easily run the system I developed without facing any difficulties in responding towards users navigations.

**iv. Consistency**

C# Multimedia-Training Package navigation maintain its consistencies. That includes the consistency of page design plus the flash movies as introduction of the system. These allow users to easily navigate and maintain their mood of learning C# through my system.

**v. Package Contents**

Contents of my Training Package are technically complete and interactive that meet my main purpose – users able to understand the C# coding from its root to its leaf node.

## 7.3 System Limitations

**i. Inconsistent Voice**

Voice used in the system package was recorded manually. As a result, there are flaws on my voice consistency. The depth and speed of voices are different. However, I managed to maintain my slang of language.

**ii. Jargon Words**

Based on the subject contents of the system package – C# programming language, it is difficult to explain in everyday English form. Opposed to my proposal earlier – for those without any programming background will face difficulties in understanding my system package. Especially when C# is the latest language that inherits the technology of C, C++, Visual Basic and Java.

iii. **User without speaker and soundcard could not use my training package**

Since I explained everything in voice form, it is impossible for the user who does not have soundcard and speaker to enjoy my training of the multimedia package.

## 7.4 Future Enhancement

The package could be further enhanced in the following way:

A) For the system package itself:

- The help system could be fully documented as well as enable the user to search the help system.
- The searching system could be inserted where the user could search for a particular detail on all topics.
- The animation could be further improved with the insertion of various ActiveX control.
- A database should be used to store the various questions where the user could be tested on a random basis. This means that user could be tested on different questions each time they enter the tutorial section.
- Since the package is divided into subtopics, those subtopics files could be easily integrated with the Internet but the size of files should be reduced before the integration could be made.

B) Marketing Purposes

- Convert it into shockwave html extension to allow online marketing where the user able to access the package from the Internet.

## 7.5 Gained Knowledge and Experience

Knowledge and experience has been gained throughout the development of this project. Among the are:

- Experience in learning a new programming languages outside the formal classroom.

- Enhanced creativity and initiative in developing the most appropriate system for end-user.
- Sharpened problem-solving skill by finding solution to all problems faced during the development of the project.
- Practicing proper time management to divide the time appropriately among all the subjects taken, and not neglecting any of them.
- Improved confidence in own ability.
- Willingness to listen and evaluate other opinions and correct oneself.

## 7.6 Summary of Chapter 7

The development of this system is not just to fulfill the requirements for a degree, but also as an experience for future use. This will be helpful when I have to face with the task of developing another multimedia system. The exercise is hoped to inspire confidence in the student capability.

Even though this multimedia training package of C# was developed without applying much complicated coding, I still need to present the latest language that not all people know about it, and it takes longer period for me to master C# besides preparing on how to present it. Moreover, it needs high sense of creativity to fulfill all the requirements of a multimedia system. Most of my designing timeframe was spent on choosing appropriate graphics, music and recording voice in order to obtain an attractive system interface to the end-user. Their needs and desires were taken into considerations in making sure that this system would be beneficial towards them.

To conclude, the C# Interactive Multimedia-Training Package was successfully developed within the given timeframe. Most of the functions named in the analysis and design were incorporated into the system. It is hopes that this system will be fulfilling the needs of the end-user and help them in coping with fast flow of language technology.

# Appendix

# Appendix A: The Author of .NET Training Course – DR. BERTRAND MEYER

**Dr. Bertrand Meyer** is one of the pioneers of modern software and object technology. He is the creator of the Eiffel programming language and the author of *Object-Oriented Software Construction*. He has been working with Microsoft's .NET platform and development team for more than a year prior to .NET official release, culminating in a joint appearance with Bill Gates at the Microsoft Professional Development Conference, where .NET was unveiled.

# Appendix B: Sample Of Survey Form

Name: *Johriah Norfizlina binti Ismail*

Age: *23*

Profession: *Student of Computer Science*

1. Have you heard of C# programming language?

■ Yes
☐ No

2. If Yes, From where?

■ Magazines, Newspapers or other printed sources
☐ Friends
☐ Internet, TV or other electronic devices
☐ Others, please specify

.....

3. Have you used any multimedia-training package before?

☐ Yes
■ No

4. If Yes, what is your comments on that multimedia training package?

.....

.....

If No, what is your expectation from multimedia-training package?

*I would expect a good interface design with good color combinations and suitable concepts. A very much movie-like will be more appreciated by me. I prefer a good professional color and graphics to be applied. Navigation should be the first priority.*

Appendix C: User Manual

# Fizlin's C# Interactive Course

## Errata

This page contains tips, user requirements and overview of C# interactive training course.

**Tips for running C# Interactive Course.**

- It is advisable to close all the applications that running in your Windows whenever you wish to start with this C# Interactive Course.
- Once you have run the CD-ROM, C# Interactive Course will be running its' .exe file automatically in your Windows.

**User Requirements.**

- Windows 9x, 2000, Millennium.
- 32MB RAM
- Color Monitor with 256-color or greater Video Card
- CD-ROM driver
- Sound card
- A set of speaker

Plus

- Best viewed on 800x600 resolution.

**Overview of C# Interactive Course Contents**

These are contents of C# Interactive Course:

- Lesson 1: A Simple Welcome Program
- Lesson 2: Expressions, Types and Variables
- Lesson 3: Control Statements – Selections
- Lesson 4: Control Statements – Loops
- Lesson 5: Methods

- Lesson 6: Namespaces
- Lesson 7: Introduction to Classes
- Lesson 8: Class Inheritance
- Lesson 9: Polymorphism
- Lesson 10: Properties
- Lesson 11: Indexers

**Running the codes of C# Interactive Course**

In order to compile and run the codes of C# that was inserted into this training course; you may need a .NET Framework. This framework is included in the CD-ROM - .NET Framework beta version. . NET framework compatible with Windows NT and above. This is what needed to be done in installing the framework:

1. Click on NET folder.
2. Here, you will find an NET Framework SDK Installer.
3. Click the Installer to install it.
4. You need to find suitable and desire path to install the framework.
5. Once done, you may type the codes in a notepad and save it in a folder. (You may also find the Lecture codes in C#_Codes folder)
6. Open Windows in DOS mode.
7. Type in your programs name – csc <file_name>.cs //note that csc stands for C# compiler, <file_name> is your filename (*make sure you are in the folder where the program you desire to run and compile located).
8. Type <file_name> - to run your program once the compilation is successful.

Hint: The latest version of Visual C++ has the features on running and compiling C# codes.

*Note: .NET Framework is only compatible with Windows NT and above (Millennium, 2000, XP)*

# References