

DCCT*1020: Computer Programming I

<http://snowwhite.cis.uoguelph.ca/~qmahmoud> [select Teaching]

MOTIVATION and INTRODUCTION

1. What is Computer Science?	2
2. What is Programming?	2
3. What is Distributed Computing?	2
3.1 Distributed Computing is Good	3
3.2 Distributed Computing is Hard	3
4. Course Topics	4
5. Pair Programming	4
5.1 What is Pair Programming?	4
5.2 Benefits of Pair Programming	4
5.3 Tips	5
6. The Internet	5
6.1 Overview of the Internet	5
6.2 The Web	6
6.3 The Client/Server Model	6
6.4 Web Resource	7
7. HTML	7
7.1 Tools	8
7.2 Anatomy of an HTML Document	8
7.3 HTML Elements	9
7.4 Styles and Sizes	11
7.5 Lists	13
7.6 Other Tags	14
7.7 Tags with Attributes	15
7.8 Images	16
7.9 Linking	16
7.10 Special Characters	17
7.11 HTML Validation	17
Appendix A: Color names and RGB values	18
Appendix B: HTML Tutorials Online	18

1. What is Computer Science?

The study of interesting things we can do with computers. The Applied Computing degree at the University of Guelph-Humber is very applied, with concentration on Distributed Computing.

Some of the applied topics you will study during your four years include:

- Programming
- Algorithms
- Software engineering
- Processor organization and hardware
- User interface design
- Operating systems
- Networking
- Distributed programming
- Wireless networking
- Network Security
- Distributed Programming
- And more distributed programming

2. What is Programming?

If you haven't programmed before, you will find programming a very challenging thing to learn. The problems you will face initially are related to concepts and language syntax.

Programming is really the art of writing instructions for a machine to execute. The most important thing to realize is that **computers don't solve problems they execute solutions.**

[more on programming in class]

3. What is Distributed Computing?

The recent emergence of the Internet and the availability of cheaper and more powerful computers are enabling a new generation of computer application (known as "network" or "distributed" applications) to be built.

These applications consist of components distributed across the network. These components work together to coordinate activities or share resources (hardware, software, and data). The World Wide Web (or Web for short) represents a good example of a distributed system.

3.1 Distributed Computing is Good

A number of key characteristics are responsible for the usefulness of distributed computing applications:

- Multi-user applications (email)
- Resource sharing (office workers can share the same expensive color printer)
- Scalability (problems of any size can be solved)
- Availability (if one computer is not available, another can be used)
- Transparency (the system is perceived as a whole – local and remote resources can be accessed using identical operations – example: local vs. remote printer)

3.2 Distributed Computing is Hard

The bad news is that the above characteristics are *not automatic consequences* of distributed computing. The design and implementation of distributed computing applications offer new challenges when compared to the construction of traditional single-computer applications. Some of the issues that need to be dealt with include: multiple failure modes, security issues, use of multiple technologies, testing and debugging.

The Applied Computing program at the University of Guelph-Humber is intended to equip graduates with the knowledge and skills required to contribute to this vital area of information technology.

The market for distributed computing applications is enormous and growing (especially with wireless computing). Distributed computing applications can be found in almost every business sector including: Banking, Manufacturing, Entertainment, Healthcare, Education, Transportation...

4. Course Topics

This course assumes no background in computer programming. We start with simple concepts but move quickly. This course will cover topics related to:

- Designing simple web sites with HTML
- Developing simple applications using JavaScript
- Designing and developing simple applications using the Java programming language

5. Pair Programming

In this course, I will be using a technique known as “Pair Programming”. This basically means that all labs and assignments will be done in pairs. This technique promotes knowledge sharing and prepares you for industrial teamwork.

5.1 What is Pair Programming?

Two programmers working side-by-side, collaborating on the same design, algorithm, code or test. One programmer, the driver, has control of the keyboard/mouse and actively implements the program. The other programmer, the observer, continuously observes the work of the driver to identify tactical (syntactic, spelling, etc.) defects and also thinks strategically about the direction of the work. On demand, the two programmers can brainstorm any challenging problem. Because the two programmers periodically switch roles, they work together as equals to develop software. [Reference: <http://www.pairprogramming.com>].



5.2 Benefits of Pair Programming

In Pair Programming, pairs of developers write all production code. This is *not as inefficient as it may sound*. In fact, pair programming provides many benefits [Reference: Extreme Programming: Ken Auer and Roy Miller]:

- All design decisions involve at least two brains
- At least two people are familiar with every part of the system
- Changing pairs spreads knowledge, tips, and tricks throughout the team
- Code is always being reviewed by at least one person

5.3 Tips

Some people may not like the idea of pairing because:

- Want to be a hero by developing some wonderful code: Code may include features not needed
- Afraid of exposing deficiencies: Check part of your ego at the door and be fearless at the personal level
- Don't wish to waste partner's time (can't explain what I am trying to achieve): Investigate along and then get back together and talk about the conclusions. Don't use a disagreement with your partner as an excuse to work alone.
- I am in a bad mood (bad trip on subway or spilled coffee down my shirt): Your teammates didn't arrange your subway trip.

INTRODUCTION

6. The Internet

This section provides an overview of the Internet and the components of the World Wide Web (or Web for short), and then discusses the Web markup language (HTML).

6.1 Overview of the Internet

The Internet is a network of networks. In other words, all local area networks on university campuses, companies, etc are connected together to

form the global Internet. The Internet is based on the client/server model – the server is a program that offers services, and the client is the requester of these services.

The computers that are connected to the Internet are heterogeneous – they are from different vendors and run different operating systems. For example, there are Unix machines, IBM-compatible machines, Windows machines, MacOS machines, PalmOS handheld devices, and even Java-enabled cellular phones. In order for all these machines and devices to communicate together, they must all speak the same language. This language (or set of rules) is known as a protocol.

Currently, many protocols are used on the Internet, including:

- FTP (File Transfer Protocol): This is used to transfer files between machines connected to the Internet
- SMTP (Simple Mail Transfer Protocol): This is used to transfer email messages from the sender to the recipient
- HTTP (Hyper Text Transfer Protocol): This is the protocol used on the Web.

6.2 The Web

The Web was invented in 1989 at CERN, the European Particle Physics Laboratory. Physicists at CERN developed an authoring language and distribution system for creating and sharing multimedia-enabled, integrated electronic documents over the Internet. The authoring system consisted of a markup language known as HTML (or the Hyper Text Markup Language) and a browser for browsing multimedia documents. With this system, no longer did authors have to distribute their work as fragmented collections of pictures, sounds, and text. HTML unified those elements. Moreover, it enabled hypertext linking, whereby documents automatically reference other documents, located anywhere around the world.

Lift-off happened when the NCSA (National Center for Supercomputing Applications) at the University of Illinois, Urbana-Champaign released a web browser called *Mosaic*, which was an instant success.

6.3 The Client/Server Model

The Web is the most widely used Client/Server system. The server is the Web, or HTTP, program that is constantly running and listening for requests,

and a Web browser is known as a client. The communication is based on request-reply message (the client sends a request and the server sends back a reply). All request and reply messages adhere to the HTTP specification.

6.4 Web Resource

Anything accessible via the Web is known as a resource. Examples are HTML documents and images. The standard way to locate resources on the Web is using the Uniform Resource Locator (URL). A URL identifies the **protocol**, **host**, **optional port number**, and **document path**. A URL has the following format:

<http://www.guelphhumber.ca:80/index.html>

- The protocol can be: http, https (secure http used for e-commerce), ftp, telnet, etc.
- The host is the machine running the Web, or HTTP, server.
- The port number uniquely identifies the server among other servers running on the same machine. The default port number for HTTP is 80. If the HTTP server is running on port 80, no need to specify it in the URL. If, however, the HTTP server is listening on a different port number then you must specify it.
- The path is the location of the resources requested

NOTE: It is important to note that a URL cannot contain spaces or characters such as comma (,), semicolon (;), ampersand (&), ..etc.

7. HTML

HTML stands for the Hyper Text Markup Language. It is a markup (or a document-layout) language that has been derived from the Standard Generalized Markup Language (SGML). This is different from a programming language, which is used to perform actions in a designated order. In HTML, text is markup with *tags* (keywords contained in pairs of angle brackets). You will study many of the commonly used tags in HTML later in this document and next week's document.

HTML has gone through at least four iterations in barely 8 years. The current version is 4.0.x. These notes describe version 4.0.1, however, some

deprecated (in transition period) HTML will also be discussed. More on this in class.

IMPORTANT NOTE: In this course I will not be covering everything there is to know about HTML. I will only cover the most widely used HTML stuff, which allow you to create cool Web sites.

7.1 Tools

In order to compose HTML documents you need two tools: a text editor and a browser. You can use the barest of barebones text editors to create HTML documents. Example of a text editor is “notepad” (for Windows machines). You also need a Web browser (such as Netscape Communicator and Microsoft Internet Explorer) to test and refine your documents.

NOTE: Many Web site designers use word processors that are capable of saving your documents in HTML format and WYSIWYG (what-you-see-is-what-you-get) HTML composers such as FrontPage. In this course, however, in order to learn HTML, I would like you to use Notepad or a similar editor with no HTML capabilities.

7.2 Anatomy of an HTML Document

An HTML document consists of text that defines the content, and tags that are used to define the appearance of the document. The structure of an HTML document is as follows:

- An HTML document must have <html> as its root. This tells the browser that the entire document is composed of HTML.
- It has a head (<head>) and a body (<body>). You put information about the document (such as the title) in the head, and the content you want to display in the browser inside the body.

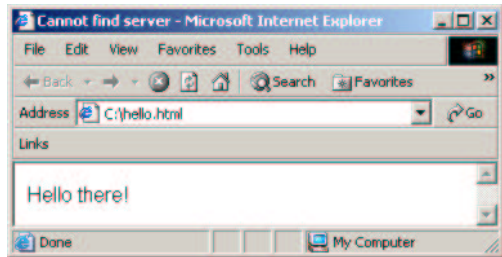
Here is an example: hello.html

```
<html>
<head>
<title>My First HTML Document</title>
</head>
<body>
```

Hello there!

```
</body>
</html>
```

The `body` is where you want to place the content of your document. This may include text, images, tables, frames, forms....etc. If you load the file `hello.html` in a browser, you'd see something like this:



7.3 HTML Elements

This section provides some of the HTML elements and the corresponding tags that are used frequently in HTML documents:

- **Headings:** there are six heading tags `<h1>` through `<h6>`, with corresponding end tags `</h1>` through `</h6>`. The browser displays the content of the headings in very large (`<h1>`) to very small (`<h6>`) font sizes. The text inside `<h4>` is usually the same size as regular text. Note that the heading tags break the current text flow, standing alone on lines even though there might be no line break before or after a heading tag.
- **Paragraphs:** the `<p>` tag is used to start a new paragraph. It is important to note that the `<p>` tag is one of the few tags that do not require an end tag. However, it is a good idea to always have an end tag for every start tag (more on this in class).
- **Line-breaks:** the `
` tag is used to break a line of text and place subsequent characters on the next line.
- **Divisions:** similar to the `
` tag, the `<div>` tag is used to break a line of text and place subsequent characters on the next line. This is not as widely used as `<p>` or `
`.
- **Horizontal rules:** the `<hr>` tag is used to provide a horizontal rule. When a browser encounters an `<hr>` tag, it breaks the flow of text and draws a line completely across the display window on a new line.
- **Preformatted text:** sometimes you may want the browser to display a block of text as-is. In such cases, the `<pre>` tag can be used.

- **Comments:** if you wish to add comments (which are ignored by the browser) to your HTML documents, use the `<!-- this is my comment -->`

Here is an example (`second.html`) demonstrating all of these tags:

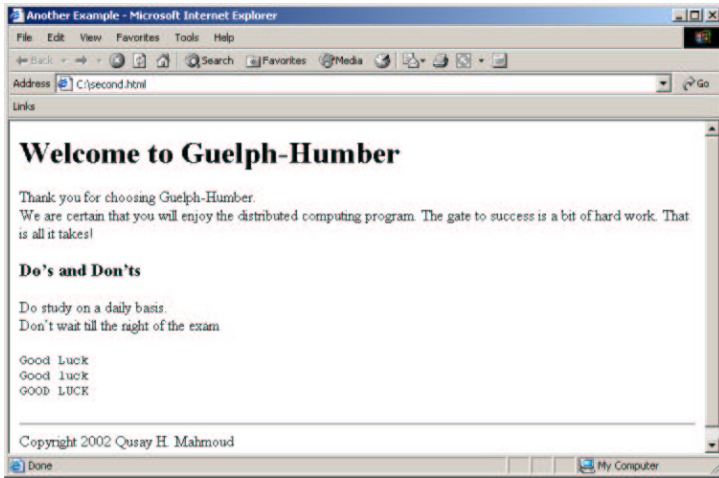
```
<html>
<head>
<title>Another Example</title>
</HEAD>
<body>
<!-- this is my second example -->
<h1> Welcome to Guelph-Humber</h1>

<p>
Thank you for choosing Guelph-Humber.<br>
We are certain that you will enjoy the distributed
computing program. The gate to success is a bit of hard
work. That is all it takes!

<h3>Do's and Don'ts</h3>
Do study on a daily basis.<BR>
Don't wait till the night of the exam

<p>
<PRE>
Good Luck
Good luck
GOOD LUCK
</PRE>
<hr>
Copyright 2002 Qusay H. Mahmoud
</BODY>
</HTML>
```

When you load `second.html` in a browser, you see something similar to this:



Here are a few things to note about the above example (second.html):

- HTML documents are saved in file with extension .html or .htm
- HTML tags are not case sensitive (while it is a matter of style, it is recommended that you write all your tags in lowercase letters).
- HTML doesn't care about spaces and carriage returns unless these are inside a `<pre>` tag. In other words, text with tags is displayed in lines with as many words as will fit in each line.
- The browser ignores HTML comments

7.4 Styles and Sizes

- Blockquotes: the tag `<blockquote>` can be used to set a block of text off from the normal flow. Most browsers indent the text. [The reason I put this tag here because I'd like to show you how I usually use it. Use your imaginations when creating HTML documents!]
- The font style and size can be set with the following tags:
 - Boldface: ``
 - Italics: `<i>`
 - Underline: `<u>`

Here is an example (third.html):

```

<html>
<head>

<title>Third Example</title>

</head>

<body>

<h2>Styles and Fonts</h2>

<p>

HTML is quite easy to learn as you can see. Not all
tags require an end tag, but it is a good idea to
<i>always</i> have an <u>end tag</u> for <b>every</b>
start tag. :-)

<p>

<blockquote>

<hr>

<b>Note</b>: HTML is not case sensitive. This means
that you can write your HTML tags using lowercase or
uppercase letters...or even a mix of both. It is however
recommended that you use lowercase letter for tags.

<hr>

</blockquote>

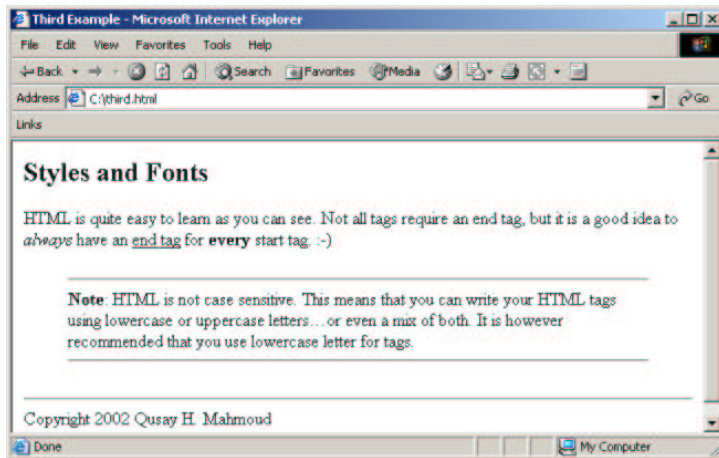
<hr>

Copyright 2002 Qusay H. Mahmoud

</body></html>

```

If you load third.html in a browser, you will see something like this:



7.5 Lists

Lists can be:

- Ordered: use `` for ordered lists (items are preceded by a sequence value).
- Unordered: use `` for unordered lists.

List items are added, in ordered and unordered lists, using the `` tag. Note that lists can be nested.

Here is an example (nba.html):

```
<html>
<head>
<title>List Example</title>
</head>
<body>
<h2>NBA Teams</h2>
<ul>
  <li>Los Angeles Lakers
    <ol>
      <li>Name A
      <li>Name b
    </ol>
  <li>Chicago Bulls
```

```
    <li>Miami Heat
  <li>Altanta Hawks
</ul>
<h2>NBA Rankings</h2>
<ol>
  <li>Chicago Bulls
  <li>Los Angeles Lakers
  <li>Atlanta Hawks
    <ol>
      <li>Name A
      <li>Name B
      <li>Name C
    </ol>
  <li>Miami Heat
</ul>

</body>
</html>
```

When loading nba.html in a browser, you see something like this:

NBA Teams

- Los Angeles Lakers
 1. Name A
 2. name B
- Chicago Bulls
- Miami Heat
- Altanta Hawks

NBA Rankings

1. Chicago Bulls
2. Los Angeles Lakers
3. Atlanta Hawks
 1. Name A
 2. Name B
 3. Name C
4. Miami Heat

7.6 Other Tags

Some other cool tags include:

- Text can be set into superscript and subscript using `<sup>` and `<sub>`, respectively. For example:

- The number: `2³` = 8
- The number: `8₂` = 1000
- Text can be struck out with `struck out text` or with `<strike>struck out text</strike>`
 - SALE. Was: `<strike>$12.99</strike>`. Now: \$4.95.
- You can use `` and ``. Depending on the browser, `` might be equivalent to `` and `` might be equivalent to `<i>`.
- The `<small>` and `<big>` tags make it easy to decrease and increase the size of text, respectively.
- The `<blink>` tag can be used to blink text on and off.
- The `<center>` tag can be used to center text, graphics, tables, and so on.

7.7 Tags with Attributes

So far we have seen tags with no attributes. Some of the tags we have discussed do take optional attributes:

- The `<hr>` tag, for example, have the following optional attributes, which can be combined:
 - The `size` attribute: the default is two or three pixels. You can use the `size` attribute to thicken the rules
 - `<hr size=12>`
 - `<hr size=32>`
 - The `width` attribute: the default rule is drawn across the full width of the view window. You can shorten or lengthen rules with this attribute:
 - `<hr size=30 width=30>`
 - `<hr size=20 width="75%">`
 - The `align` attribute: the default alignment is center. But you can use this attribute to set alignment to one of three values: left, right, center.
 - `<hr size=20 width="50%" align=left>`
 - `<hr align=right size=20 width="75%">`
- The `<body>` tag has the following optional attributes (which have been deprecated in HTML 4.0. Now style sheets are used)
 - The `background` attribute: refers to an image anywhere.
 - `<body background="http://aol.com/file.gif">`
 - `<body background="./me.gif">`
 - The `text` attribute: sets the foreground color of text
 - `<body text="blue">`

- `<body text="#0000ff">`
- The `link` attribute: sets the color of text marking unvisited hypertext links
 - `<body text="red" link="yellow">`
- The `vlink` attribute: sets the color of text marking visited hypertext links
- The `alink` attribute: sets the color of text marking hypertext links when selected by the user.
- The heading tags (`<h1>` through `<h6>`) take an `align` attribute that may have one of the values: left, right, center. The default alignment is left.

7.8 Images

Images are inserted into a document using the `` tag. This tag has one required attribute and several optional ones:

- The `src` attribute is required. Its value is the image file or url.
 - ``
 - ``
- The `align` attribute is optional. It can have one of the values: left, right, middle, and some others...
 - ``
- The `width` and `height` attributes can be used to set the width and height of the image (make it smaller or larger):
 - ``
- The `alt` attribute is used to provide an alternative for browsers that do not support graphics or have images turned off:
 - ``

7.9 Linking

To create a hypertext link (or hyperlink) to another place in the same document or to another document, use the `<a>` anchor tag:

- The `<a>` tag takes an `href` attribute, which is used to specify the URL of the target of a hyperlink. The value of the `href` attribute is any valid document URL, absolute or relative:
 - `go to java`
 - `some info`
 - ``
- You can use the `name` and `id` attributes with the `<a>` tag to create a fragment identifier within a document. The link to a document fragment must be preceded by a pound sign (#):

- o `<h1>Hello</h1>`. To link to this fragment use: `go to hello`
- o `<h1 id="hello">Hello`. To link to this fragment use: `go to hello`. You can also link to a fragment of another document using:
`go to hello`

7.10 Special Characters

Some characters have special meanings in HTML and therefore when used in a document, the browser may get confused. Some of these characters are: Less than (<), greater than (>), ampersand (&). To display these characters, use their code names:

To display <, use <

To display >, use >

To display ©, use ©

A list of all these is available at:

http://hotwired.lycos.com/webmonkey/reference/special_characters

7.11 HTML Validation

Making errors in a conventional programming language produces a fatal error, preventing the program from running. Making errors in HTML is not fatal – the browser will make its best effort to render the page, but will probably not display the page as you intended.

There are several HTML validation services available online, which validate your HTML documents against the HTML specification and making sure you have followed all the rules. A good validation service is available at:
<http://validator.w3.org/file-upload.html>












When using such a service, you need to specify the document type, which specifies which version of HTML is used in the document. Here is an example:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

More on this in class.

Next week I will cover: HTML Tables, HTML Frames, and HTML Forms.

Appendix A: Color names and RGB values

 Black = "#000000"	 Green = "#008000"
 Silver = "#C0C0C0"	 Lime = "#00FF00"
 Gray = "#808080"	 Olive = "#808000"
 White = "#FFFFFF"	 Yellow = "#FFFF00"
 Maroon = "#800000"	 Navy = "#000080"
 Red = "#FF0000"	 Blue = "#0000FF"
 Purple = "#800080"	 Teal = "#008080"
 Fuchsia = "#FF00FF"	 Aqua = "#00FFFF"

Appendix B: HTML Tutorials Online

There are many good HTML tutorials and other reference material available online:

- A Beginner's Guide to HTML (This is all you need really)
<http://archive.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerPrintable.html>
- HTML Tutorial: <http://www.w3schools.com/html>
- HTML 4.0.1 Specification: <http://www.w3.org/TR/html4>
 This one is ONLY for the curious. You do not need to read it.