*Alan Bridgeman*

*Date*

*AccessiblITY Education*

Proposed Approaches for the Inclusion of Accessibility in Computer Science Curriculum at University of Manitoba

Table of Contents

[1. Introduction 7](#_Toc457378860)

[2. Overview 7](#_Toc457378861)

[3. Methodology 7](#_Toc457378862)

[4. Models 8](#_Toc457378863)

[1. Single Course Approach 8](#_Toc457378864)

[2. Integrated Approach 8](#_Toc457378865)

[Single Course Approach 9](#_Toc457378866)

[Course Outline 9](#_Toc457378867)

[Topic Outlines 11](#_Toc457378868)

[Diversity and Society (1) 11](#_Toc457378869)

[What is diversity? (1.1) 11](#_Toc457378870)

[What is a disability? (1.2) 11](#_Toc457378871)

[Inclusion in society (1.3) 11](#_Toc457378872)

[Inclusion in politics and law (1.3.1) 12](#_Toc457378873)

[Overview of Rights and Freedoms (1.3.1.1) 12](#_Toc457378874)

[Overview of legislation (1.3.1.2) 12](#_Toc457378875)

[The shift to be inclusive (1.3.2) 13](#_Toc457378876)

[Accessible/Alternative formats (2) 14](#_Toc457378877)

[Conveying information in multiple formats (2.1) 14](#_Toc457378878)

[Alternative format documents (2.1.1) 14](#_Toc457378879)

[Braille and large print documents (2.1.1.1) 14](#_Toc457378880)

[Plain text documents (2.1.1.2) 14](#_Toc457378881)

[Properly tagged documents (2.1.1.3) 15](#_Toc457378882)

[Assistive devices and how people use them (2.2) 15](#_Toc457378883)

[Screen readers (2.2.1) 15](#_Toc457378884)

[Text-to-speech software (2.2.2) 15](#_Toc457378885)

[Magnification software (2.2.3) 16](#_Toc457378886)

[Speech-to-text software (2.2.4) 16](#_Toc457378887)

[Technologies that assist (2.3) 16](#_Toc457378888)

[Captioning (2.3.1) 16](#_Toc457378889)

[Descriptive video (2.3.2) 16](#_Toc457378890)

[Semantics and verbosity (3) 17](#_Toc457378891)

[What it means to be semantic (3.1) 17](#_Toc457378892)

[The semantic web (3.1.1) 17](#_Toc457378893)

[Understanding through semantics (3.1.2) 17](#_Toc457378894)

[Levels of verbosity (3.2) 18](#_Toc457378895)

[Non-descriptive (3.2.1) 18](#_Toc457378896)

[Generic and vague (3.2.2) 18](#_Toc457378897)

[Accurate (3.2.3) 18](#_Toc457378898)

[Overly detailed (3.2.4) 18](#_Toc457378899)

[Screen reader perception (3.3) 19](#_Toc457378900)

[Languages, APIs and techniques (4) 19](#_Toc457378901)

[WAI-ARIA (4.1) 19](#_Toc457378902)

[Java Accessibility API (4.2) 19](#_Toc457378903)

[Generation of alternate format documents programmatically (4.3) 20](#_Toc457378904)

[Standards and regulations (5) 20](#_Toc457378905)

[Laws and legislation (5.1) 20](#_Toc457378906)

[AMA ICT standard (5.1.1) 20](#_Toc457378907)

[AODA ICT standard (5.1.2) 20](#_Toc457378908)

[Section 508 (U.S.) (5.1.3) 21](#_Toc457378909)

[Technical standards (5.2) 21](#_Toc457378910)

[WCAG (5.2.1) 21](#_Toc457378911)

[ATAG (5.2.2) 21](#_Toc457378912)

[UAAG (5.2.3) 22](#_Toc457378913)

[Testing (6) 22](#_Toc457378914)

[Practices (6.1) 22](#_Toc457378915)

[Tools (6.2) 22](#_Toc457378916)

[WAVE (6.2.1) 22](#_Toc457378917)

[Inclusive design (7) 23](#_Toc457378918)

[General Principles (7.1) 23](#_Toc457378919)

[Creating flexible customizable interfaces (7.2) 23](#_Toc457378920)

[Adding Accessibility to legacy systems (8) 23](#_Toc457378921)

[Considerations (8.1) 23](#_Toc457378922)

[Redesign (8.2) 24](#_Toc457378923)

[Conclusions (9) 24](#_Toc457378924)

[The responsibility of developers as it relates to diversity (9.1) 24](#_Toc457378925)

[About the role diversity plays in software design (9.2) 24](#_Toc457378926)

[Integrated Approach 25](#_Toc457378927)

[Undergraduate Courses 25](#_Toc457378928)

[COMP 3010 - Distributed Computing 25](#_Toc457378929)

[COMP 3020 - Human-Computer Interaction 1 25](#_Toc457378930)

[COMP 3040 - Technical Communication in Computer Science 25](#_Toc457378931)

[COMP 3350 - Software Engineering 1 26](#_Toc457378932)

[COMP 4020 - Human-Computer Interaction 2 26](#_Toc457378933)

[COMP 4050 - Project Management 26](#_Toc457378934)

[COMP 4060 - Topics in Computer Science 26](#_Toc457378935)

[COMP 4350 - Software Engineering 2 27](#_Toc457378936)

[COMP 4620 - Professional Practice in Computer Science 27](#_Toc457378937)

[Graduate Courses 27](#_Toc457378938)

[COMP 7570 - Advanced Topics in Computer Science 1 27](#_Toc457378939)

[COMP 7920 - Advanced Topics in Graphics and Human Interfaces 27](#_Toc457378940)

[Look at how to properly implement inclusive design looking at specific elements like how to have an interface that seamlessly moves from no-style to style, from inverted colours to standard colours and to have adjustable text size among many other metrics. 28](#_Toc457378941)

[Appendix A: Glossary 29](#_Toc457378942)

[Appendix B: Career Postings 30](#_Toc457378943)

[Microsoft 30](#_Toc457378944)

[Accessibility Program Manager 30](#_Toc457378945)

[Senior Technical Evangelist 31](#_Toc457378946)

[Google 33](#_Toc457378947)

[Software Engineer, Accessibility 33](#_Toc457378948)

[Facebook 34](#_Toc457378949)

[Front End Engineer - Accessibility, Intern/Co-op 34](#_Toc457378950)

# Abstract

The Accessibility for Manitobans Act (AMA)[[1]](#footnote-1) became law on December 5, 2013 making improving accessibility for people living with disabilities law in Manitoba. The standards for

was Given the incoming legislation for AMA and the potential for a national legislation[[2]](#footnote-2), it is becoming increasingly important that developers as a minimum requirement have a working knowledge of the standards and guidelines they will be held up to in regards to accessibility. Given the lack of high quality, available training and education in the area of accessibility in the Computer Science curriculum at University of Manitoba and if willing perhaps the knowledge of how to go further to better promote diversity and inclusivity in the programs we (as students and alumni) create. I think it is best said: “Computers and software aid, inform, and enrich our lives. A degree in Computer Science will help you contribute to the incredible ways computers are shaping our lives.”[[3]](#footnote-3)

If the Computer Science department is willing to extend the resources to integrate these topics into the curriculum, using either of the two proposed approaches, then the quality of graduates from the program will improve as they will understand and will consider accessibility when developing software.

# Introduction

As a fourth year Computer Science undergraduate student at the University of Manitoba living with a visual impairment, I have relied on assistive devices and software like magnifiers, screen readers, keyboard input and many other such devices to achieve the same experience as that of a person without a disability on a day-to-day basis my entire life. As such, I feel compelled to use my skills to improve or create similar or better tools for the next generation of living with disabilities or the growing aged population so that they may have the same or better opportunity than I have been given.

To this end, I’ve subsequently taken it upon myself to investigate the opportunity available to me enter this type of career path; much of what I’ve found has included a mere subsection or brief summary of the requisite knowledge required to enter this type of career.

# Methodology

Most of my knowledge I’ve gained through reading articles, joining mailing lists and attending seminars over the past couple of years. I’ve used my own knowledge along with consultations with industry experts to create two models of how accessible technology concepts might be integrated into the Computer Science programs here at the University of Manitoba.

The industry professionals I’ve consulted are Lisa Snider, Senior Accessibility Consultant/Owner, Everything Accessibility[[4]](#footnote-4), Jeff Buhse, Assistive Technologist, Student Accessibility Services[[5]](#footnote-5), University of Manitoba, and Jutta Treviranus[[6]](#footnote-6), Inclusive Design Research Centre (IDRC)[[7]](#footnote-7), Ontario College of Art and Design University (OCADU).

One noteworthy area of study I discovered through my research is that of Biomedical Engineering (BME) which is, in some senses, the physical aspect of what I am attempting to endeavor upon. That is to say, that this area looks at making devices and prosthetics that help rehabilitate people through advanced electronics and engineering. While in the physical world this provides a lot of promise in the increasingly digital era we live in this area provides a meager means for better sources of input and output rather than improving the interaction itself (through things like semantics, interface redesign and dynamic output). Thinking this sounds a lot like Human Computer Interaction (HCI)? Well in some regards I think it is, but in contrast to more traditional HCI practices of design I think it talks more about how to communicate information in multiple formats and in different mediums. In short, how to make more flexible programs and websites so that users can use them in their own “special” way rather than a single design or set of perspectives set out by the designer. This includes creating audible and keyboard navigable interfaces, caption or transcript systems for videos and many other examples.

# Summary of Proposed Approaches

## Single Course Approach

The first model for integrating accessible technology concepts into the computer science program involves creating a single course that takes the students through their accessibility obligations as software developers; the proper practices and techniques to employ when creating technologies that could be used by persons living with disabilities; and many, many other topics related to creating accessible technologies.

## Integrated Approach

The second method, a distributed approach, inserts accessibility related content into pre-existing courses as applicable. This allows students to have a broader sense of the impact that this topic has as it relates to different subjects. It is taught in conjunction with other technologies and is therefore more ingrained in what the student understands about the language or technologies they’ve learned.

# Approaches

## Single Course Approach

Creating an individual course allows students to explore accessibility and its implications in software development in a complete and in-depth way allowing the subject matter to extradite itself from getting lost in the shuffle as it might be in the more integrated approach. Further to this point, it allows full exploration of the standards and technologies used to create technologies for persons living with disabilities perhaps even fostering a curiosity in some students leading them to pursue a graduate degree here at the University of Manitoba. Potential drawbacks I see are: it would be costly with little guarantee of attendance from students especially initially; very few local experts or people who know the subject matter well enough and there is very little if any accreditation or certification for students or the class in general.

### Course Outline

1. [Diversity and Society (1 ⅓ - 2 ⅙ hours)](#_Diversity_and_Society)
   1. [What is diversity (15 - 30 minutes)](#_What_is_diversity)
   2. [What is a disability (15 - 30 minutes)](#_What_is_a)
   3. [Inclusion in society (50 minutes - 1 ⅙ hour)](#_Inclusion_in_society)
      1. [Inclusion in politics and law (45 minutes - 1 hour)](#_Inclusion_in_politics)
         1. [Overview of Rights and Freedoms (30 minutes)](#_Overview_of_Rights)
         2. [Overview of legislation (15 - 30 minutes)](#_Overview_of_legislation)
      2. [The shift to be inclusive (5 - 10 minutes)](#_The_shift_to)
2. [Accessible/Alternative formats (1 hour 25 minutes - 2 hour 5 minutes)](#_Accessible/Alternative_formats_(2))
   1. [Conveying information in multiple formats (50 minutes - 1 hour)](#_Conveying_information_in)
      1. [Alternative format documents (50 minutes - 1 hour)](#_Alternative_format_documents)
         1. [Braille and large print documents (15 minutes)](#_Braille_and_large)
         2. [Plain text documents (5 - 15 minutes)](#_Plain_text_documents)
         3. [Properly tagged documents (30 minutes)](#_Properly_tagged_documents)
   2. [Assistive devices and how people use them (25 minutes - 45 minutes)](#_Assistive_devices_and)
      1. [Screen readers (10 - 15 minutes)](#_Screen_readers_(2.2.1))
      2. [Text-to-speech software (5 - 10 minutes)](#_Text-to-speech_software_(2.2.2))
      3. [Magnification software (5 - 10 minutes)](#_Magnification_software_(2.2.3))
      4. [Speech-to-text software (5 - 10 minutes)](#_Speech-to-text_software_(2.2.4))
   3. [Technologies that assist (10 - 20 minutes)](#_Technologies_that_assist)
      1. [Captioning (5 - 10 minutes)](#_Captioning_(2.3.1))
      2. [Descriptive video (5 - 10 minutes)](#_Descriptive_video_(2.3.2))
3. [Semantics and verbosity (4 - 9 hours)](#_Semantics_and_verbosity)
   1. [What it means to be semantic (2 - 5 hours)](#_What_it_means)
      1. [The semantic web (1 - 3 hours)](#_The_semantic_web)
      2. [Understanding through semantics (1 - 2 hours)](#_Understanding_through_semantics)
   2. [Levels of verbosity (1 - 2 hours)](#_Screen_reader_perception)
      1. [Non-descriptive (15 - 30 minutes)](#_Non-descriptive_(3.2.1))
      2. [Generic and vague (15 - 30 minutes)](#_Generic_and_vague)
      3. [Accurate (15 - 30 minutes)](#_Accurate_(3.2.3))
      4. [Overly detailed (15 - 30 minutes)](#_Overly_detailed_and)
   3. [Screen reader perception (1 - 2 hours)](#_Screen_reader_perception)
4. [Languages , APIs and techniques (8 - 14 hours)](#_Languages,_APIs_and)
   1. [WAI-ARIA (5 - 8 hours)](#_WAI-ARIA_(4.1))
   2. [Java Accessibility API (2 - 5 hours)](#_Java_Accessibility_API)
   3. [Generation of alternate format documents programmatically (1 - 2 hours)](#_Generation_of_alternate)
5. [Standards and regulations (4 - 9 hours)](#_Standards_and_regulations)
   1. [Laws and legislation (1 ½ - 3 hours)](#_Laws_and_legislation)
      1. [AMA ICT standard (30 minutes - 1 hour)](#_AMA_ICT_standard)
      2. [AODA ICT standard (30 minutes - 1 hour)](#_AODA_ICT_standard)
      3. [Section 508 (U.S.) (30 minutes - 1 hour)](#_Section_508_(U.S.))
   2. [Technical standards (2 ½ hours - 6 hours)](#_Technical_standards_(5.2))
      1. [WCAG (1 - 3 hours)](#_WCAG_(5.2.1))
      2. [ATAG (1 - 2 hour)](#_ATAG_(5.2.2))
      3. [UAAG (30 minutes - 1 hour)](#_UAAG_(5.2.3))
6. [Testing (1 ¼ hours - 2 ½ hours)](#_Testing_(6))
   1. [Practices (1 - 2 hours)](#_Practices_(6.1))
   2. [Tools (15 - 30 minutes)](#_Tools_(6.2))
      1. [WAVE (15 - 30 minutes)](#_WAVE_(6.2.1))
7. [Inclusive design (1 ½ - 3 hours)](#_Inclusive_design_(7))
   1. [General Principles (1 - 2 hours)](#_General_Principles_(7.1))
   2. [Creating flexible customizable interfaces (30 minutes - 1 hour)](#_Creating_flexible_customizable)
8. [Adding Accessibility to legacy systems (1 ½ hours - 3 hours)](#_Adding_Accessibility_to)
   1. [Considerations (30 minutes - 1 hour)](#_Considerations_(8.1))
   2. [Redesign (1 - 2 hours)](#_Redesign_(8.2))
9. [Conclusions (30 minutes - 1 hour)](#_Conclusions_(9))
   1. [The responsibility of developers as it relates to diversity (15 - 30 minutes)](#_The_responsibility_of)
   2. [About the role diversity plays in software design (15 - 30 minutes)](#_About_the_role)

Total: 23 ⅔ hours - 45 ¾ hours

If a regular course is approximately 33 ⅓ hours long and the average for the timeline proposed is approximately 34 ¾ hours I feel that this is within a reasonable margin of error

### Topic Outlines

#### Diversity and Society (1)

Before you can talk about inclusive/accessible/assistive technology you have to understand the reason making these “special” technologies is relevant and you need to understand the users who would be using these sorts of technologies as to aid in the best design practices.

**Timeline**: 1 ⅓ - 2 ⅙ hours

#### What is diversity? (1.1)

A good place to start to build a foundation for this course is talking about what diversity is. In particular what diversity in IT means and the design and process implications that it can have. Also in this portion of the course topics like the business cases for why having accessible IT is economically a good idea are are also covered.

**Timeline**: 15 - 30 minutes

#### What is a disability? (1.2)

While the previous portion of the course should focus on the implications and the business case for diversity. This portion is meant to understand the particular needs of the user base who would be using the technologies being created in or from this course. This is things like talking about the division between permanent vs temporary disabilities and the division between physical, mental and learning disabilities and the different needs each of these groups in regards to their various IT devices and services.

**Timeline**: 15 - 30 minutes

#### Inclusion in society (1.3)

The intent of this portion is to understand the extent to which society has begun to put emphasis on creating an accessible or inclusive environment in general and the many different things that are currently out there to promote this sort of movement or societal push.

**Timeline**: 50 minutes - 1 ⅙ hour

#### Inclusion in politics and law (1.3.1)

The discussion of legislation and law comes up a few times throughout this course in this particular occurrence it is to do a quick overview simply to understand that politicians are putting stock into this and forcing organizations and persons to make this a common consideration and to understand what this means.

**Timeline**: 45 minutes - 1 hour

#### Overview of Rights and Freedoms (1.3.1.1)

Perhaps before diving into the current and somewhat more relevant legislation and laws it makes sense to do a brief discussion on the Human Rights Code, UN Rights and Freedoms of Persons with Disabilities, and several others.

**Related Resources**:

* <http://web2.gov.mb.ca/laws/statutes/ccsm/h175e.php>,
* <http://laws-lois.justice.gc.ca/eng/acts/H-6/>,
* <http://www.un.org/disabilities/convention/conventionfull.shtml>

**Timeline**: 30 minutes

#### Overview of legislation (1.3.1.2)

Overview of current disability related legislation like the Accessibility for Manitobans Act (AMA), the Accessibility for Ontarians with Disabilities Act (AODA), Americans with Disabilities Act (ADA), Rehabilitation Act (Rehab Act) and the Australian Disability Discrimination Act. Showing there is relevant content in this legislation without necessarily diving into the details.

NOTE: I'm saving the full review of the actual ICT standard till later for effect/understanding.

**Related Resources**:

* <http://accessibilitymb.ca/pdf/accessibility_for_manitobans_act.pdf>
* <https://www.ontario.ca/laws/statute/05a11>
* <https://www.ada.gov/pubs/adastatute08.htm>
* <http://www.usbr.gov/cro/pdfsplus/rehabact.pdf>
* <https://www.legislation.gov.au/Details/C2016C00763>

**Timeline**: 15 - 30 minutes

### The shift to be inclusive (1.3.2)

A brief discussion on the movement towards even more legislation and the growing global will to have this sort of law or enforced standards

**Related Resources**:

* <http://www.esdc.gc.ca/en/consultations/disability/legislation/index.page>

**Timeline**: 5 - 10 minutes

### Accessible/Alternative formats (2)

Being acquainted with this movement towards more inclusive or accessible environments and technology. It’s time to start to dive into the next layer down which is creating accessible and alternate format materials. This will allow us to better understand how to develop software that can output in helpful and proper formats.

**Timeline**: 1 hour 25 minutes - 2 hour 5 minutes

### Conveying information in multiple formats (2.1)

A critical point throughout the course will be to communicate things through multiple formats as much as reasonable. In this portion the intent is to explore some of the multiple alternative formats to attempt to generate later on in the course.

**Timeline**: 50 minutes - 1 hour

### Alternative format documents (2.1.1)

A particular area of interest is documents and the different kind of documents that you can generate in particular the tagged documents will nuance well into topic 3.

**Timeline**: 50 minutes - 1 hour

### Braille and large print documents (2.1.1.1)

Explore what makes up a Braille and large print document and the considerations and issues that arise when trying to generate them particularly mechanically or manually as a discussion of programmatic generation will happen later.

**Timeline**: 15 minutes

### Plain text documents (2.1.1.2)

A very brief discussion about why generating a purely plain text document with no formatting information can be a good idea for accessibility and how you might manually translate a document with no formatting into a plain text document. This also helps students fully understand the role to which graphics and styling play in a practical manner.

**Timeline**: 5 - 15 minutes

### Properly tagged documents (2.1.1.3)

This is a particularly important sub-section given that it introduces this idea of semantics and communicating through more than just content. This will focus on how to generate documents (PDFs, Word Docs, etc.) with proper formatting and tags somewhat manually (using Office, Acrobat, etc.).

**Timeline**: 30 minutes

### Assistive devices and how people use them (2.2)

This section is to be about the devices/software that users with disabilities might use. As such our programs would have to interface with them meaning that we should be aware of them.

**Timeline**: 25 minutes - 45 minutes

### Screen readers (2.2.1)

A screen reader allows somebody is unable to read a screen to be able to have what’s on the screen announced this has a lot of nuances that are commonly overlooked and therefore given that this course will be extensively diving into this technology I’ve assigned a longer time frame.

**Timeline**: 10 - 15 minutes

### Text-to-speech software (2.2.2)

Quite similar to the screen reader this classification of software usually used for software that will only announce text as “requested” rather than attempting to announce the entirety of the screen this type of software is commonly used by persons living with disabilities and other persons who may want just the text audibly. Some examples include: Dragon

**Timeline**: 5 - 10 minutes

### Magnification software (2.2.3)

Magnifications software is, as it sounds, software that magnifies the screen typically for those persons living with a visual impairment. Some examples includes: ZoomText

**Timeline**: 5 - 10 minutes

### Speech-to-text software (2.2.4)

Also sometimes called dictation software this software has become a widespread technology for an evolving hands-free market

**Timeline**: 5 - 10 minutes

### Technologies that assist (2.3)

So far in topic two we’ve covered conveying information in multiple ways particularly alternate formats of documents and some devices and software that actually help persons with disabilities in this section the idea is to talk about the technologies and techniques we as content developers can use to actually promote accessibility of content

**Timeline**: 10 - 20 minutes

### Captioning (2.3.1)

Talk about the type of captioning there is and some of technologies out there to help captioning videos talk briefly about some of the things you should insure you do when captioning a video, etc…

**Timeline**: 5 - 10 minutes

### Descriptive video (2.3.2)

The idea of providing a second video that has a layered audio track that describes visual ques and happenings on the screen. This technology is predominantly for persons living with visual impairments.

**Timeline**: 5 - 10 minutes

### Semantics and verbosity (3)

In this third topic we’ll begin to explore the software developer’s role to communicate information through non visual information in the source code.

**References**: [2.1.1.3](#_Properly_tagged_documents)

**Timeline**: 4 - 9 hours

### What it means to be semantic (3.1)

In this portion of the topic the idea is to talk about what it means to be semantic and how it relates to the way in which we communicate between technologies in this particular course we are somewhat focusing on how to communicate with assistive devices like screen readers but this is still a fairly generic topic at this point.

**References**: [2.2.1](#_Screen_readers_(2.2.1))

**Timeline**: 2 - 5 hours

### The semantic web (3.1.1)

The “semantic web” isn’t actually a assistive technology concept however it provides some very good grounding for students to begin to understand the ways in which communicating through the source code rather than purely content works and the things that should and shouldn’t be done using this technique.

**Related Resources**:

* <http://www.w3.org/standards/semanticweb/>

**Timeline**: 1 - 3 hours

### Understanding through semantics (3.1.2)

Closely related to the previous section this section will attempt to get students to think more about how when working with a diverse set of technologies semantics may play in to understanding and may briefly talk about some things that maybe were not sure how to communicate through semantics yet or can only be communicated through content

**Timeline**: 1 - 2 hours

### Levels of verbosity (3.2)

The idea behind this portion of the course is to understand further the amount to which you should try to communicate through semantics both in general but also in individual attributes. I think this would be an appropriate time to talk about Alt Text.

**Timeline**: 1 - 2 hours

### Non-descriptive (3.2.1)

In this section examples of how semantics in particular things like ALT Text can exist but not say anything about what its actually trying to communicate. A common thing that may fit under here is the tendency for web content developers to have links with text like ‘click here’ which is useless given no positional idea of where the link is in relation to anything.

**Timeline**: 15 - 30 minutes

### Generic and vague (3.2.2)

This section explores examples that aren’t entirely useless but might have some more description but still don’t say the right amount and don’t communicate what the content creator wants. Something that should come up here particularly in the context of Alt Text is do we actually need it.

**Timeline**: 15 - 30 minutes

### Accurate (3.2.3)

This section is to give actually good examples of Alt Text and other semantically translated information.

**Timeline**: 15 - 30 minutes

### Overly detailed (3.2.4)

This section will talk about what happens when something like Alt Text is too detailed and simply boring. In this section a discussion about Facebook’s new auto generated Alt Text feature may be applicable given that it tends to have long Alt Text’s that don’t actually describe what the picture is actually of.

**Timeline**: 15 - 30 minutes

### Screen reader perception (3.3)

Now that semantics and verbosity have been covered in-depth its important to solidify the relationship between this and what this course is about of particular interest is screen readers given that they need to

**Timeline**: 1 - 2 hours

### Languages, APIs and techniques (4)

Finally, the good stuff this topic is about actually programming for these technologies introduce a few of the languages and APIs that help bridge the gap between assistive technologies and mainstream technologies like the web.

**Timeline**: 8 - 14 hours

### WAI-ARIA (4.1)

The Web Accessibility Initiative – Accessibility Rich Internet Application language is an extension to the already existing HTML language that provides semantics for screen readers to efficiently and effectively read and operate web pages. This includes doing things like changing content via JavaScript or altering the default reading order so that when the screen reader is going through the page it is easy to figure out what is what and where it is. Keyboard navigation and tab order can also be set with this language and lots of other things.

**Timeline**: 5 - 8 hours

### Java Accessibility API (4.2)

My experience with this API is less than than with WAI-ARIA however from what I’ve done with it it has similar features and abilities but does it in a somewhat different way customized for the java framework.

**Timeline**: 2 - 5 hours

### Generation of alternate format documents programmatically (4.3)

My thought for the portion is to talk about things like media queries that will help when a user is using something like a Braille display or some other kind of custom device where you might need to have a slightly different style arrangement. It also may talk about ways in which to pipeline from an application into a program to generate an accessible format document like Braille or large print.

**Timeline**: 1 - 2 hours

### Standards and regulations (5)

Now that students have a good handle on the technologies involved with creating accessible IT products we can go back and take a more in-depth look at the standards and regulations that would apply to students as software or IT developers.

**Timeline**: 4 - 9 hours

### Laws and legislation (5.1)

The reasons I fell this portion should come before the technical standards is because it gives context to those standards. When we cover them the idea is that the standards are no longer abstract documents that someone just decided to follow they are things that students and IT professionals have to follow as its inscribed in law and legislation.

**Timeline**: 1 ½ - 3 hours

### AMA ICT standard (5.1.1)

Being in Manitoba I felt that presenting the Manitoba legislation first made the most sense and would capture student’s attention the best.

**Timeline**: 30 minutes - 1 hour

### AODA ICT standard (5.1.2)

In a very similar vain I felt that looking at a different but similar standard in Ontario might present a nice compare and contrast in the Canadian context.

**Timeline**: 30 minutes - 1 hour

### Section 508 (U.S.) (5.1.3)

To take this compare and contrast idea even further this section is intended to give a vague idea that these types of laws and legislations exist globally and that there are differences and similarities in between the standards and the way their enforced. This section may also include a discussion about Section 508[[8]](#footnote-8) and the Harvard case that I would say is a current affairs issue that makes this type of subject to students seemingly more relevant.

**Related Resource**:

* National Association of the Deaf (“NAD”) versus Harvard University, et al. <http://creeclaw.org/wp-content/uploads/2016/02/2016-02-09-50-Report-and-Rec-re-MTD.pdf> (PDF)

**Timeline**: 30 minutes - 1 hour

### Technical standards (5.2)

Now that we’ve given context to the standards let’s talk about the actual contents of the technical standards.

**Timeline**: 2 ½ hours - 6 hours

### WCAG (5.2.1)

The Web Content Accessibility Guidelines are a set of guidelines used to dictate how web page content should be used particularly to optimize for use by persons with accessibility needs.

**Timeline**: 1 - 3 hours

### ATAG (5.2.2)

The Authoring Tool Accessibility Guidelines give an idea of how to create authoring tools that promote accessibility and allow for a user to follow content standards like WCAG.

**Timeline**: 1 - 2 hour

### UAAG (5.2.3)

The User Agent Accessibility Guidelines dictate how a user agent like a browser can best interpret accessibility related languages and APIs like WAI-ARIA.

**Timeline**: 30 minutes - 1 hour

### Testing (6)

Now that we know how to create the software its important to discuss how we can test and ensure that the software we have built works in the way in which we want it to.

**Timeline**: 1 ¼ hours - 2 ½ hours

### Practices (6.1)

Testing software is something that is in its nature very context sensitive and as such there are certain principals in which to always, sometimes or never consider when working towards accessible technology.

**Timeline**: 1 - 2 hours

### Tools (6.2)

While its great to talk about principals and studies its also important to talk about ways in which we can make the process more efficient through good useful tools that help us do our job but don’t compromise integrity.

**Timeline**: 15 - 30 minutes

### WAVE (6.2.1)

This web based tool helps evaluate web accessibility it a major way while you can’t be 100% reliant on this tool it does provide a good mechanism for a first line of defense for testing for accessibility.

**Timeline**: 15 - 30 minutes

### Inclusive design (7)

We’ve talked about how you make things accessible and how you test for accessibility but in this topic were going to talk about how to design for accessibility and the ways in which that can be done.

**Timeline**: 1 ½ - 3 hours

### General Principles (7.1)

In design the idea of guiding principals is a common theme and is something that that tells the designer what to consider and in what order to consider them and in this regards it is actually one of the most critical pieces to a good design is a good set of these guiding design principals.

**Timeline**: 1 - 2 hours

### Creating flexible customizable interfaces (7.2)

One of the particularly hard parts of inclusive design in opposition to that of typical design is this idea of “one-size-fits-one” which is to say that an interface should be able to be customizable while keeping the designers intent.

**Timeline**: 30 minutes - 1 hour

### Adding Accessibility to legacy systems (8)

Its awesome to have the opportunity to design from the ground up however reality is that this is in the scheme of things a relatively small percentage of what accessibility technology professionals do. Most of their work revolves around taking existing systems and attempting to take the elements of inclusive design and the pieces we’ve learned earlier in the course and insert them into a pre-existing or legacy system which requires close considerations to redesign and keeping the balance between certain functionality and accessibility.

**Timeline**: 1 ½ hours - 3 hours

### Considerations (8.1)

The considerations that an accessible technology professional must make when looking at a system already in place is very lengthy list and of particular interest is the considerations of what functionality need be changed or updated what functionality need be added and what functionality need be completely redesigned.

**Timeline**: 30 minutes - 1 hour

### Redesign (8.2)

While feature changes and additions are a big piece of the consideration stage the redesigning a feature entirely to make it more accessible is by far the hardest option. However given there are situations in which this arise it is appropriate to speak to it in this course and the strategies and techniques one can employ to try to do the redesign in a logical and efficient manner.

**Timeline**: 1 - 2 hours

### Conclusions (9)

In this final topic the idea is to end of on conclusions or summarizations of the th9ings in which students have learned ove4r the time of the course and the learning’s the student should take away from the course.

**Timeline**: 30 minutes - 1 hour

### The responsibility of developers as it relates to diversity (9.1)

This portion of the course is to summarize and reemphasize the responsibilities a developer has to making technology accessible to those persons living with disabilities and the public in general. It should briefly speak to the legislation and laws, the business case, the technical standards and several other similar topics.

**Timeline**: 15 - 30 minutes

### About the role diversity plays in software design (9.2)

This final portion is intended to summarize and reemphasize the considerations, practices, technologies and techniques used to make software more accessible both being designed initially or at the redesign stage.

**Timeline**: 15 - 30 minutes

## Integrated Approach

This method is a much more distributed approach and inserts accessibility related content into pre-existing courses as applicable. This allows students to have a broader sense of the impact that this topic has as it relates to different subjects. Moreover it is taught in conjunction with other technologies and is therefore more ingrained in what the student understands about the language or technologies they’ve learned. The difficulty being that most of these course are already “jam packed” with content and have barely enough time to cover the content already inscribed in their curriculum.

### Undergraduate Courses

#### COMP 3010 - Distributed Computing

An introduction to the development of client server and peer-to-peer systems through web applications, distributed programming models, and distributed algorithms.

##### Topics to include

Web Accessibility (WCAG, WAI-ARIA)

### COMP 3020 - Human-Computer Interaction 1

Human-computer interaction: human factors and usability, user-centered design, prototyping, usability evaluation.

#### Topics to include

Brief mentions and overview of inclusive design

### COMP 3040 - Technical Communication in Computer Science

This course is designed to help students become more effective and confident writers in the context of the computing profession. Students will be introduced to a broad range of written and oral presentation styles used in the computing workplace.

#### Topics to include

Alternative format documents such as Braille, Large print and web-based documents.

### COMP 3350 - Software Engineering 1

Introduction to software engineering. Software life cycle models, system and software requirements analysis, specifications, software design, testing and maintenance, software quality.

#### Topics to include

Brief mentions of accessibility and that when building software should be something to keep in mind.

### COMP 4020 - Human-Computer Interaction 2

Advanced issues in the field of human-computer interaction. Topics will be selected from current research and development issues in the field of HCI.

#### Topics to include:

Principals of inclusive design and thought given to how to make accessible interfaces.

### COMP 4050 - Project Management

Introduction to the issues involved in managing large, complex software projects.

#### Topics to include:

How to conduct a self “accessibility audit” in terms of making sure that projects you mange are accessible

### COMP 4060 - Topics in Computer Science

This course will examine topics of interest at the fourth-year level. Specific topics will vary from year to year. Topics will be selected from current research in computer science.

#### Topics to include:

I feel like accessible or assistive technology as a topic here feels apt.

### COMP 4350 - Software Engineering 2

Advanced treatment of software development methods. Topics will be selected from requirements gathering, design methodologies, prototyping, software verification and validation.

#### Topics to include:

Perhaps a more detailed discussion of how an accessibility consultant or team may play a role in the development of a software product and the validation they may perform to meet standards.

### COMP 4620 - Professional Practice in Computer Science

Background and rationale to view Computer Science in a professional context. Examination of professional ethics, intellectual property, and privacy considerations important to Computer Scientists.

#### Topics to include:

The impact diversity and accessibility has to the field of computer science in particular the legal and implied obligations a modern software developer faces. This includes the AMA, AODA, Section 508.

## Graduate Courses

### COMP 7570 - Advanced Topics in Computer Science 1

Topics of current research interest in areas of computer science, available on an occasional basis, subject to the interests and availability of faculty.

#### Topics to include:

I particularly thinking looking at topics such as accessible graphics like the SVG modification for screen readers or ChartML for charts would be interesting topics that will require a certain amount of technical knowledge such as graphics and background accessibility knowledge like what I’ve suggested in the undergraduate section as well as understanding of UX design to use these things properly.

### COMP 7920 - Advanced Topics in Graphics and Human Interfaces

Topics of interest in advanced Human-Computer Interaction (HCI). Possible topics include quantitative and qualitative evaluations, crowdsourcing methods and applications, personal informatics, persuasive technologies, technologies for special populations, and information visualization.

# Appendix A: Hierarchy and Glossary

## Hierarchy

heirarchyused in this paper

## Glossary

# Appendix B: Career Postings

To provide a brief idea of what some of the career aspirations of someone looking at going into the field of accessible/assistive technology might be able to get.

## Microsoft

As one of the most pervasive developers of a modern age particularly in the business market it is a sure telltale sign that when these positions exist in a company like this it is only a matter of time before many more developers come to create these types of positions. Moreover, Microsoft which employee many computer scientists is a leader at adopting their obligations to accessibility had has engaged in much of the discussion of how technology can move towards being more accessible.

### Accessibility Program Manager

Accessibility is a key part of Microsoft’s mission to empower people and organizations to achieve more. Our team partners across the company to make the vision a reality for people with disabilities. As part of this focus we are looking for an Accessibility Program Manager to join the Accessibility team reporting to the Chief Accessibility Officer (CAO)

Are you passionate about the customer experience of Microsoft products for customers with disabilities? Are you a mature, technical, experienced professional capable of leading initiatives to improve accessibility across our products and services? Do you possess strong organizational agility and demonstrate success in engaging executives and influencing others. Then read on!

We are looking for a technical leader to partner with the engineering teams that will drive for improved customer experience across Microsoft products and services that will positively affect the adoption and perception of our products and drive customer loyalty. They will have a wide range of skills and understanding for all aspects of disability, including blindness, mobility, deafness, cognitive and speech and how accessibility can drive improvements to the overall user experience.

#### Primary Responsibilities:

* Identify the top problem areas or issues impacting the customer experience for customers with disabilities and drive appropriate changes with engineering that will have a broad impact on the customers’ perception of Microsoft Accessibility product quality. Primary focus is on C&E engineering team.
* Be recognized and act as technical subject matter expert for Accessibility, supporting the products/services within your focus.
* Effectively exercise and demonstrate deep cross group ownership of complex projects and processes
* Develop and maintain a diverse and global perspective ensuring partnership across audience (consumer, commercial etc.) as well as the field.
* Partner with other parts of the Accessibility community across Microsoft to nail the right support experience for customers with disabilities, partnering with peers across the tema, and company to help drive accessibility efforts.

#### General Knowledge, Skill and Experience Requirements:

* Passion for technology, Accessibility and overall quality of Microsoft’s products/services.
* Passion for empathy for all customers and partners and for resolving the difficulties they face while using our products.
* Proven experience working and partnering directly with communities of consumers with disabilities and driving change into products and services.
* Demonstrated ability to develop and execute against a long-term vision of success, while ensuring successful completion of short-term objectives.
* Ability to represent Microsoft and the complexities of Accessibility to technical and non-technical audiences – including executive, IC and team audiences from varying skil sets.
* Deep functional, hands-on expertise and experience with Accessibility features and products
* Strong written, verbal, and presentation skills.
* Team player with proven ability to build trustful relationships and to work in diverse teams.
* Sound problem resolution and decision making skills.
* Understanding of the Microsoft product offerings withing Developer and C&E
* Proven ability to effectively collaborate and manage multiple groups and tasks to schedule
* The individual selected for the position should be able to work independently with minimal supervision.
* Solid understanding of software architectures, software development life cycle, and Microsoft products are essential.
* Demonstrated experience with Accessibility standards and guidelines such as 508, 255,CVAA and WCAG.
* Deep understanding of Accessibility trends and Assistive Technologies.

### Senior Technical Evangelist

Accessibility is a key part of Microsoft’s mission to empower people and organizations to achieve more. Our team partners across the company to make that vision a reality for people with disabilities. As part of this focus we are looking for a Senior Technical Evangelist with experience and passion for accessibility to join the team. Reporting under the Chief Accessibility Officer (CAO), this role will lead our efforts to provide a voice for people with al abilities. If you are passionate to improve the user experience for people with disabilities and a mature, technical, experienced professional capable of bu9ilding trusted relationships with customers and partners, the read on!

Trusted relationships with NGOs and Assistive Technology (AT) providers are critical to further Microsoft’s commitment to empower every person on the plenty to achieve more. This role offers the opportunity to improve relationships with some of our most prominent connections points with our customers and AT providers and gain feedback around best practices within product accessibility and usability.

As an Accessibility Technical Evangelist your primary responsibility will be to be the voice on accessibility and listen and tell our Accessibility story to customers, NGOs, ATs and employees at customer sites, events , and company locations. You will empower internal teams, especially the field, with information and expertise on accessibility of Microsoft’s products and services interfacing with the product and engineering teams internally

The ideal candidate will have experience in client-facing roles and have demonstrated success leading technical and economic value discussions with executives and senior technical decision makers, driving key decisions and implementation strategies. Candidates need to possess the ability to impact and influence stakeholders with a high degree of autonomy, energy, flexibility and the drive to create real and measurable business results. Technical knowledge of accessibility is key to the success of the role and it is crucial to have the ability to demo technical solutions that will benefit people with disabilities.

#### Responsibilities of this role include:

* Build working relationship with ATs, NGO, and MSFT Field in region to build our Accessibility awareness and enable Field readiness.
* Ability to use your expertise to guide others’ work, establish and share accessibility standards and policies.
* Work closely with other accessibility, design and usability specialists within Microsoft engineering teams to enable success
* Develop and maintain multipl3e listening and communication channels, business and adoption metrics and other relevant indicators for success
* Evangelize our product and services across audiences and help enable adoption supporting the Field with your accessibility and product expertise.

#### Qualifications:

* Able to successfully communicate and demonstrate technical ideas to technical and non-technical audiences
* Be a trusted advisor and credible in delivering the Microsoft vision for accessibility
* Passionate about and deep understanding of accessibility technologies as well as business drivers and emerging trends
* Ability to build technical and business credibility with senior executives and grow mind share with highly strategic customers and partners
* Track record of driving decisions collaboratively and and resolving conflicts with consistent and reliable follow through
* Exceptional verbal and written communication skills
* Ability to connect technology with measurable business value
* Demonstrate technical thought leadership in customer-facing situations
* Expertise in a wide range of assistive technology including screen readers, screen magnifiers, speech input, high contrast and the full range of options on Microsoft platforms.
* Solid understanding of the competitive landscape for accessibility
* Experience working with the assistive technology industry to support development in their products
* Detailed knowledge of accessibility guidelines (such as Section 508 and WCAG 2.0) and technologies (such as ARIA)

#### Requirements:

* BS in computer science or equivalent work experience
* Minimum 6 years of experience working on a consumer or enterprise product
* Experience presenting accessibility issues and solutions to leadership
* Experience shipping and iterating on successful products
* Adaptability, flexibility and the ability to do the right thing under pressure and tight deadlines
* Ability to implement both strategy and tactical needs of the business

## Google

### Software Engineer, Accessibility

Google aspires to be an organization that reflects the globally diverse audience that our products and technology serve. We believe that in addition to hiring the best talent, a diversity of perspectives, ideas and cultures leads to the creation of better products and services. This team will allow you to work on projects that will aim to make all our products accessible to those in need.

Google’s software engineers develop the next-generation technologies and change how billions of users connect, explore, and interact with information and one another. Our products need to handle information at massive scale, and extend well beyond web search. We’re looking for engineers who bring fresh ideas from all areas, including information retrieval, distributed computing, large-scale system design, networking and data storage, security, artificial intelligence, natural language processing, UI design and mobile; the list goes on and is growing every day. As a software engineer, you will work on a specific project critical to Google’s needs with opportunities to switch teams and projects as you and our fast-paced business grow and evolve. We need our engineers to be versatile, display leadership qualities and be enthusiastic to tackle new problems across the full-stack as we continue to push technology forward.

Google is and always will be an engineering company. We hire people with a broad set of technical skills who are ready to tackle some of technology’s greatest challenges and make an impact on millions, if not billions, of users. At Google, engineers not only revolutionize search, they routinely work on massive scalability and storage solutions, large-scale applications and entirely new platforms for developers around the world. From AdWords to Chrome, Android to YouTube, Social to Local, Google engineers are changing the world one technological achievement after another.

#### Responsibilities:

Design, develop, test, deploy, maintain and improve software. Manage individual project priorities, deadlines and deliverables.

#### Qualifications:

##### Minimum qualifications:

* BS degree in Computer Science, similar technical field of study or equivalent practical experience.
* 1 year of experience working in accessibility technology.
* Experience working with two or more from the following: web application development, Unix/Linux environments, mobile application development, distributed and parallel systems, machine learning, information retrieval, natural language processing, networking, developing large software systems, and/or security software development.
* Software development experience in one or more general purpose programming languages. Working proficiency and communication skills in verbal and written English.

##### Preferred qualifications:

* Master’s, PhD degree, further education or experience in engineering, computer science or other technical related field.
* Experience with one or more general purpose programming languages including but not limited to Java, C/C++, C#, Objective C, Python, JavaScript, or Go.
* Interest and ability to learn other coding languages as needed.

## Facebook

### Front End Engineer - Accessibility, Intern/Co-op

Facebook was built to help people connect and share, and over the last decade our tools have played a critical part in changing how people around the world communicate with one another. With over a billion people using the service and more than fifty offices around the globe, a career at Facebook offers countless ways to make an impact in a fast growing organization.

Facebook is seeking a Front End Engineer intern to focus on the accessibility of Facebook’s various product offerings. This person would work with the Accessibility Engineering team to improve infrastructure accessibility and compatibility of Facebook’s products with assistive technologies. This position is based in our Menlo Park headquarters.

#### Responsibilities:

* Directly improve the accessibility of Facebook’s products
* Work closely with product teams on accessibility
* Participate in design and code reviews
* Interact with other teams and team members to incorporate their innovations and vice versa

#### Requirements:

* Experience in the field of accessibility and knowledge or strong interest of assistive technologies (JAWS, VoiceOver, Dragon, NaturallySpeaking) and their interactions with web technologies (HTML, CSS, JS, ARIA)
* Ability to write high-performance, reusable code for UI components
* Investigate and desire to untangle complex, subtle problems
* Demonstrated software engineer/UIE experience via an internship, work experience, or coding competition
* Exceptional communication skills
* Pursuing BS or MS degree in Computer Science or a related technical field.

1. Accessibility for Manitobans Act (AMA). Retrieved from <http://www.accessibilitymb.ca> [↑](#footnote-ref-1)
2. Consulting with Canadians on planned accessibility legislation. Retrieved from http://www.esdc.gc.ca/en/consultations/disability/legislation/index.page [↑](#footnote-ref-2)
3. Department of Computer Science: Undergraduate Studies. Retrieved from http://cs.umanitoba.ca/undergraduate/ [↑](#footnote-ref-3)
4. Everything Accessibility. Retrieved from http://everythingaccessibility.com [↑](#footnote-ref-4)
5. Student Accessibility Services. Retrieved from http://umanitoba.ca/student/saa/accessibility/index.html [↑](#footnote-ref-5)
6. Brief Biography: Jutta Treviranus. Retrieved from http://idrc.ocadu.ca/about-the-idrc/staff-pages/112-jutta-treviranus-director [↑](#footnote-ref-6)
7. Inclusive Design Research Centre (IDRC). Retrieved from http://idrc.ocadu.ca [↑](#footnote-ref-7)
8. Section 508. Retrieved from https://www.section508.gov [↑](#footnote-ref-8)