DALHOUSIE UNIVERSITY
FACULTY OF COMPUTER SCIENCE

CSCI 4177/5709 — Advanced Topics in Web Development

Assignment 3

Overview

In CSCI 4177/5709, 35% of your grade involves work done for assignments. These assignments are meant to put the skills and theory you have learned in lectures and tutorials, along with the skills you had prior to joining the course, to practice. Assignments are considered to be individual deliverables that can be used towards your group project, although together they do not make up 100% of your group project. As such, it is strongly recommended for you to manage your time appropriately and use the assignments component of this course as an outlet for you to try out ideas you may be interested on using for your project. Furthermore, while **assignments** can be used towards your group project, they are meant to reflect the students' individual work, and therefore **are NOT to be carried out in groups** unless specifically indicated by the Instructor; though you may consult with group project members or classmates and TAs during lab/tutorial sessions.

There are a total of THREE (3) assignments in this course. Although, initially the assignments are not too difficult, they do get progressively harder as you learn new concepts and techniques covered in the course. As such, do keep in mind the scope of your assignments when managing your time. **Assignments are due by the END OF DAY (i.e., 11:59PM) on the date noted on BrightSpace, and must be submitted through both Brightspace and Git Lab unless otherwise specified on the assignment handout**. Finally, students should also be aware that they will be tested on topics included in each of the THREE (3) assignments, in addition to material covered in the course lectures, tutorials, and in-class discussions and activities.

It also goes without saying that any instance of **academic dishonesty** will be reported. **If you decide to use and modify any existing code, e.g., code found on online or printed sources or code used during in-class/tutorial examples, you are expected to provide author attribution with an explanation of why the piece of code is necessary for your work, where, how and why the code or section of code was modified in your submission's README.txt file, README file template is provided on the course's Resources tab on Brightspace. Further, if you use AI Tools in your work, you will also be expected to provide the prompts, responses, justifications, and customization approaches of these responses in your README file.**

Descriptions of the assignments are posted in advance so that you are aware of what is expected in each assignment, and are able to manage your time appropriately as **assignment due dates will NOT have any extensions. You are NOT expected to submit all assignments at the same time – each assignment has its own due date**.

Any late changes (if necessary) made to this document or any of the assignments will be notified in class and via email.

Purpose. The purpose of these assignments are to test your comprehension of the various concepts discussed in class, and your ability to apply them to solve a given problem.

Grades. Each deliverable will be graded out of 100 points, and will be scaled to 10, 8, and 12 points for Assignments 1, 2 and 3, respectively.

Software / Code Editors. Coded deliverables must be completed **without** the aid of "visual" website generating software. This includes desktop programs such as Dreamweaver or web based programs such as Wix. You can use tools such as Notepad++ / Vi / Vim / Sublime Text, Visual Studio Code, etc.

Submission. All deliverables must be submitted on Brightspace (**https://dal.brightspace.com**) and Git Lab (**https://git.cs.dal.ca**).

Late Submission Policy. Late assignments are **not** accepted. However, no penalty will be assessed for assignments that are late due to documented situations (See Syllabus).

Academic Integrity. Dalhousie academic integrity policy applies to all submissions in this course. You are expected to submit your own work. Please refer to and understand the academic integrity policy, available at: **http://www.dal.ca/dept/university_secretariat/academic-integrity.html**

Content for the website. Do not copy and paste content from any websites into your prototype application. You will have to create your own content to include on your website.

Assignment 3

[12% Individual Deliverable]

While Assignment 2 focused on planning the Front-End and Back-End of one of your project's features, Assignment 3 focuses on applying **Back-End development techniques, approaches, and APIs**, for defining and developing the functionality of a web application based on a given set of requirements (i.e., your A3 submission and your group's project). Specifically, in this assignment you will be developing the back-end functionality for ONE (1) of the features of your project (i.e., the feature you planned for in Assignment 2). As you will likely be coordinating with members of your group for this Assignment (e.g., a feature you are developing may affect a feature being developed by a teammate), you are expected to write clean and self-descriptive code with meaningful variable, object, function and class name that allow for any incoming developer to quickly understand your code. In addition to sourcing your code, you are also encouraged to provide any descriptions you see fit within your Assignment 3's README file (e.g., the Front-End features these Back-End functions/methods/properties/APIs provide functionality for) and only use brief descriptive comments in your code when strictly necessary.

If you decide to use and modify any existing code, e.g., code found on online or printed sources, or code used during in-class/tutorial examples, you are expected to provide author attribution in your README.txt file providing an explanation of why the piece of code is necessary for your work, where, how and why the code or section of code was modified. You are encouraged to use the README template available on Brightspace, as this is an annotated template meant to guide you in this process. Keep in mind that simply stating "code was modified" does not provide sufficient information required in your programming assignments, **the amount of detail expected in your README file is illustrated in the README template. Further, if you use AI Tools in your work, you will also be expected to provide the prompts, responses, justifications, and customization approaches of these responses in your README file.**

Finally, it goes without saying that any instance of **academic dishonesty** will be reported. If you decide to use and modify any existing code, e.g., code found on online or printed sources or code used during in-class/tutorial examples, you are expected to provide author attribution in your code comments **as well as an explanation of when, how and why your code or section of code was modified**.

Learning Objectives:

1. To assess and apply suitable Back-End APIs and Frameworks for the purpose of developing a prototype application, given a set of proposed guidelines (e.g., wireframes, devices, expected functionality).

2. To work in teams and choose best-solutions proposed by each team member in order to create one cohesive design for a prototyped web application.

Requirements:

For your Assignment 3, you must do the following:

A3.1. As part of Assignment 3, **EACH member of your group is expected to use a single single Git Lab repository, where each individual student has setup individual branches. For this assignment, each of the members of your group are also expected to have a single URL for remotely accessing your group's A3 submission.** The expectation of A3 is for all group members to have their code integrated into their main project repo and deployment. **A3 submissions where the student provides a separate deployment URL (different from the group's deployment URL) or a separate A3 repo URL that is in no way connected to the group's main project repo, will receive a grade of 0,** You must include a single line comment at the top of your files to indicate the author of a file. As well, your individual README.txt file, which you will need to submit through Brightspace, must include the list of files for your A3 submission.

Note: In this assignment, each group member will submit an individual README.txt file *on BrightSpace, providing the necessary details for that individual's work (i.e., features developed along with its related tasks, files created, code referencing and justifications, integration instructions). On GITLab, your group is expected to have a single README file.* ***The markers will visit the group's application URL and check the features created by each individual student (i.e., we will revise your individual README file to identify which features each individual group member has created prior to grading). Failure to provide these details will result in a possible maximum grade of 50%. Code that is not integrated with the group's master repository will NOT be graded and thus will receive a grade of 0%.***

A3.2. **For this assignment, you will develop the ONE (1) FULL feature** from the features you chose in Assignment 2. Developing one full feature entails developing the front-end and back-end for all tasks in your feature.

Note: *You will need to* ***submit the COMPLETED back-end and front-end of the feature you have chosen for this assignment, i.e., all of the feature's related tasks are expected to be fully functional.*** *While you may develop this feature using any programming language, it is the responsibility of EACH student in a group to ensure all members coordinate which features will be completed (from the project's MUST-HAVE list) and ensure that any dependencies are properly completed (e.g., the shopping cart feature is dependent on the checkout feature, completing a checkout feature implies the shopping cart must also be completed).*

A feature is defined as a group of related task for a particular overall purpose. For example, a student choosing the User Profile Management feature implies the student is expected to plan for

the following tasks: user registration, user login, view user profile, update user profile, forgot password, user logout, delete user account. ***Failure to complete ONE (1) FULL feature will result in a possible maximum grade of 50%.***

A3.3. You **may use** any front-end or back-end off-the-shelf boilerplates or frameworks (e.g., Bootstrap, Foundation, Skeleton, jQuery, AngularJS, React.JS, View.JS, etc.).

Note: *You* ***may NOT*** *use any off-the-shelf web templates regardless of whether they are freely available or not. Additionally,* ***you are expected to significantly customize the boilerplates of frameworks you use*** *(i.e., you* ***cannot*** *simply implement them 'as-is').*

A3.4. The feature you develop must reflect the requirements specified by you in your Project Proposal, and Assignment 2 submission (e.g., sitemap, wireframes, task flow diagrams, process workflows, use cases, click streams, etc).

Note: *You* ***may NOT*** *use any off-the-shelf web templates regardless of whether they are freely available or not. Additionally, you are expected to significantly customize the boilerplates of frameworks you use (i.e., you* ***cannot*** *simply implement them 'as-is'). Additionally, it is perfectly acceptable for your Assignment 3 submission to be slightly different from the requirements specified in previous work, as you are expected to refine your project idea throughout the term.*

A3.4. You **may NOT use *Lorem Ipsum* text** for the content of your pages, any content you use must be meaningful, grammatically correct, and suitable for the web. Additionally, any forms must use meaningful labels and messages.

Note: *Your UI Design must illustrate a clearly defined content hierarchy throughout your submission. Additionally, you must strive to ensure the design of your application compliments the content of your work. You are encouraged to work towards an aesthetically pleasing website that* ***applies the design and development UI, UX, Usability, and Accessibility principles*** *discussed in class. You may use Creative Commons images and/or logos with proper author attribution (provided through code comments, and/or* ***README.txt*** *file).*

A3.5. **Your assignment must be W3C compliant**, i.e., it must pass W3C front-end validations tests (e.g., HTML and CSS).

Note: *Failure to submit valid code will result it a possible maximum grade of 50%. If your assignment does not validate due to framework-specific tags or code, these errors will be overlooked (e.g., Angular's ng-app HTML attribute). As well, any validation warnings will not affect your grade.*

A3.6. **Your files should be organized in folders within your assignment folder**. i.e., your CSS files should be inside a **'css' folder**, your included files should be inside a **'includes' folder**,

your images should be inside an **'images' folder**, your javascript files should be inside a **'js' folder**, your fonts should be inside a **'fonts' folder,** etc.

Note: *If the Framework or API you have chosen to develop your assignment with, requires a specific folder structure different from what is specified in A3.6,* ***you must include a brief explanation in your README.txt*** *file.*

A3.7. **Your assignment should apply usability principles discussed in class**, e.g., usable front-end validation and user feedback techniques to validate form fields, and provide proper error recovery messages in case a field does not validate.

Note: *Proper user feedback in forms includes the use of AJAX confirmation or success messages, as well as failure messages to the user. Your messages should also take into consideration the security of your application.*

A3.8. As it is expected for you to use this submission in your group project, and you will likely be coordinating with members of your group for this Assignment (e.g., a feature you are developing may affect a feature being developed by a teammate), you are expected to write clean and self-descriptive code with meaningful variable, object, function and/or class names that allow for any incoming developer to quickly understand your code. In addition to sourcing your code, you are also encouraged to provide any descriptions you see fit within your Assignment 3's README file (e.g., the Front-End features these Back-End functions/methods/properties/APIs provide functionality for) and only use brief descriptive comments in your code when strictly necessary.

Note: *Do keep in mind that excessive commenting is not considered an appropriate or efficient way of writing code. As such, you are encouraged to ensure your functions and variables have meaningful names, and that your comments are brief but meaningful and only used when absolutely necessary so as to minimize any performance issues from lengthy descriptions in your code.*

Marking Rubric

The following grading criteria, as shown on Table 1, will be used for marking your assignment:

TABLE 1. CSCI 4177/5709 RUBRIC FOR PROGRAMMING WORK

Dimensions	Does Not Meet Expectations	Somewhat Meets Expectations	Meets Expectations	Exceeds Expectations
Feature (15%)	Fails to meet the criteria of fully developing ONE (1) feature with all of its corresponding tasks. i.e., feature is incomplete or does not compile. **(0 - 3 points)**	Somewhat developed ONE (1) feature, but most of its corresponding tasks are missing or incomplete or do not perform as expected. **(4 - 6 points)**	Assignment submission meets the criteria of ONE (1) full feature with all of its corresponding tasks. However, 1 or 2 tasks are incomplete or do not perform as expected. **(8 - 12 points)**	Assignment submission meets the criteria of ONE (1) full feature with all of its corresponding tasks. Feature is complete and works successfully. **(14 - 15 points)**
Tasks (20%)	Fails to implement the functionality of all tasks related to a given feature. **(0 - 3 points)**	Implemented some of the front-end and/or back-end functionality of up to HALF of the expected tasks, AND / OR these tasks seem disjointed or DO NOT belong to a SINGLE feature. **(10 - 14 points)**	Implemented the front-end and/or back-end functionality of the expected tasks, AND these tasks belong to a SINGLE feature, BUT some functionality problems still exist. **(15 - 17 points)**	Assignment submission successfully implements the front-end and back-end functionality of all tasks related to a given feature. **(19 - 20 points)**
Frameworks and APIs (20%)	Fails to implement front-end frameworks and/or does not provide justification for not using a framework. **(0 - 3 points)**	Implements front-end frameworks but fails to customize its implementation. **(10 - 14 points)**	Implements front-end frameworks and somewhat customizes its implementation. **(7 - 8 points)**	Successfully implements and fully customizes front-end frameworks. **(19 - 20 points)**
Content (10%)	Fails to implement proper content hierarchy. Content is not well organized on the front-end. Poor use of Lorem Ipsum content. **(0 - 3 points)**	Content hierarchy is somewhat defined and content organization is somewhat lacking. **(4 - 6 points)**	Content hierarchy is defined. Content organization is still somewhat lacking. **(7 - 8 points)**	Successfully implements proper content hierarchy throughout. Content is well organized. **(9 - 10 points)**
Front-End and Back-End Support (20%)	Overall front-end design is cluttered, without a clear colour palette or typographic style defined. Design elements are not consistent, impacting usability. Back-end fails to support front-end functionality **(0 - 3 points)**	Overall design is somewhat cluttered, has a somewhat clear colour palette but no clear typographic style defined. Design elements are still not consistent. Back-end somewhat supports front-end functionality. **(10 - 14 points)**	Overall design is not cluttered, has a somewhat clear colour palette and typographic style defined BUT design elements are still not consistent. However, back-end DOES fully supports front-end functionality. **(15 - 17 points)**	Overall design is clean, aesthetically pleasing. Deliverable has a clear colour palette and typographic style is well defined. Design elements are consistent and enhance the usability of the application AND back-end fully supports front-end functionality. **(19 - 20 points)**
File & Folder Structure (15%)	File structure is non-existent, unorganized, or not well defined. **(0 points)**			File structure is clearly defined and organized. **(14 - 15 points)**
README.txt	Fails to include a README file, or file is empty. **(-25 points)**	README file fails to include complete and/or correct details on code sources and citations, and/or justifications missing URL. **(-15 points)**		Includes a README file with complete and correct content such as application's URL, code referencing and justifications. **(0 points)**
Code Quality	Code is not optimized or excessively redundant, causing performance issues. Excessive and/or redundant comments, variables/function names are not meaningful. **(- 25 points)**	Code is not entirely optimized, some redundancy causing few performance issues. Code not properly commented, variables and function names are somewhat meaningful but a lengthy description is still required to understand the logic. **(-15 points)**		Code is optimized, no redundancy or any performance issues. Code is clear with succinct comments only when necessary. Variables/function names are meaningful. The logic of the code is clearly defined. **(0 points)**
W3C Compliant	Fails to be cross-browser and/or cross-platform compatible and/or apply W3C guidelines where appropriate. **(-25 points)**			Deliverable is cross-browser and cross-platform compatible, and applies W3C guidelines where appropriate. **(0 points)**
Code Deployment	Marker unable to mark submission, URL missing **(-100 points)**			Marker is able to mark submission **(0 points)**

Submission Guidelines

Your assignment must be submitted through **BrightSpace**, **Git Lab and Netlify**.

To submit your work to BrightSpace:

- As part of your individual assignment submission on Brighspace, **you are expected to ONLY-submit your individual README.txt file through Brightspace's Assignment 3 dropbox.**
- **Your README file must include the list of files you created as part of your assignment, the list of features developed** along with its related tasks, any necessary code referencing and corresponding justifications, and code integration instructions.

  Note: *The markers will visit the group's application URL and check the features created by each individual student. Failure to provide these details in your README.txt file will result in a possible maximum grade of 50%. Further, code that is not integrated with the group's master repository and properly deployed through the group's deployment URL, will NOT be graded and thus will receive a grade of 0%, the markers will NOT run any code locally.*

- **Your README file must include the URL pointing to your deployed application.** As your individual Assignment 3 is meant to be integrated with your groupmates' individual Assignment 3s, each member of your group will be expected to essentially submit the same main URL for their individual Assignment 3.

  Note: *Failure to integrate your Assignment 3 work with your groupmates' individual Assignment 3 will result in an automatic grade of 0.*

- **Your README file must also include** the link to your code repository.
- Rename your README file to match naming conventions specified in the Course Syllabus (**FName_LName_README.txt**).

  Note: *Your README file should include your name, GIT repository link, deployment link, and any code references. Failure to submit a README file and/or not include a link to your deployed application will result in a grade of 0.*

- Ensure you submit your work by the **due date specified on Brightspace.**

To submit your work to Dal's Git Lab:

- First, your group must have a project repository created on Dal's GitLab, your individual submission for your Assignment 3 will be the individual branch you created for your portion of your work, i.e.,

each individual group member will have their own individual branch, as shown on Figure 1. Ensure all your assignment files are included in your project folder.

```
CSCI 4177/5709 Tutorials
    - Tutorial1
    - Tutorial2
    ....

CSCI 4177/5709 Assignments
    - Assignment1
    - Assignment2
    ...

CSCI 4177/5708 Grp-xx
    - Individual name branch
```

Figure 1. *GitLab Folder Structure Example.*

Note: *For this assignment, your group will be submitting a single GitLab repository link, along with a single application URL. Each individual student is expected to submit an individual* ***README.txt*** *through BrightSpace.*

- Setup your project folder as a private project and add the course **Teaching Assistants (TAs) and Instructor** as **'Maintainers'** to your project, using their **CS IDs.**

  Note: *The CS ID for this course are provided in our Tutorial 2 handout and module. Failure to add the course CS ID as 'Maintainer' for your work on Git Lab will result in a maximum possible grade of 50%. Make sure that your README file includes the deployment link for your group's A3 submission and your group's project repo,* ***failure to do so will result in a grade of 0.***

- While you are free to use GitHub for deployment purposes, do keep in mind that your code repository on GitLab **MUST** be used for grading purposes.

Deploying your Work:

For the purposes of this assignment, you may use **Netlify and Render, and/or any other deployment option of your choice. Your deployment solution must be agreed upon by your entire project group**. To allow for this flexibility, **your README.txt file must include the URL from which your assignment can be accessed.**

However, should you choose to use Timberlea, below are a series of instructions to help you out.

- Login to **Timberlea** at **timberlea.cs.dal.ca** **using your CS Username and CS Password**. You may use Terminal or an FTP Client (e.g., FileZilla) to connect to Timberlea.

Note: If you are using an FTP Client, you may use **sftp://timberlea.cs.dal.ca** as your hostname. If you need help logging on to Timberlea, please follow the instructions available on the CS Support website (**https://web.cs.dal.ca/~tlin/cs_support/**)

- Once logged into **Timberlea,** go into your **'public_html'** folder and, if you have not already done so, create a folder called **'csci5709'**.

  Note: All your work **must** be reside inside your '**csci5709**' folder, this folder **must be nested inside your 'public_html' folder**. If your files are not inside your **'public_html'** directory on **timberlea.cs.dal.ca**, **the markers will not be able to access your work and you will receive a grade of 0. It is the responsibility of the student to ensure their assignments are available for grading before the due date.**

- Go into your **'csci5709'** folder and create an assignment folder called **'a3'**.

  Note: You will need to create an assignment folder for each individual assignment, as well as your final project, as we go through the term (i.e., a2, a3, A3, and project).

- Place the all the files you created for this assignment inside the **'a3'** folder you created on Timberlea.

  Note: In order for your assignment files to be accessible through a browser for testing and grading, you must ensure you are using the correct file permission settings on your files and folders. On a shared server, such as Timberlea, it is recommended to **use '755' (i.e., rwxr-xr-x) on folders**, and **'644' (i.e., rw-r--r--) on individual files**. You can set your file permissions easily through an FTP client by right clicking on the file or folder you want to set specific permission settings. Depending on your FTP client, you will need to click on **'Get Info'** or **'File Permissions'**. Once on the file permissions window, you can simply enter the numeric value described above.

- Test your assignment is readily accessible and properly working. Your URL will likely include a port address. Ensure you include this URL in your **README.txt** file.

  Note: You are encouraged to check your work through the URL specified in your README.txt file, as **the Instructor and TA will not be checking any other URL**. The rule of thumb is "if you can see your assignment on a browser through your assignment's URL, the TA and Instructor can see and grade your assignment". It is the student's responsibility to ensure their submission is accessible and working as expected.

- Using Development Frameworks:

  - If as part of your assignment you plan to use a development framework such as Node or Angular, do keep in ming that you will have to use a custom port when launching your web application. Ports 1000 through 40000 are allowed through the firewall for this purpose.

Note: Most students should be able to use their CS ID. However, if you do encounter issues with your account, please stop by the FCS Help Desk located on the main level of the Goldberg Computer Science building.

- If as part of your assignment you plan to use CodeIgnitor, a PHP development framework, you may simply download these files into your public_html directory and serve them from your Timberlea account. CodeIgnitor also includes a database configuration file, so you may need to have your own copy of this file.

Note: Should you have any issues, please stop by the FCS Help Desk located on the main level of the Goldberg Computer Science building.

- If as part of your assignment you plan for use .NET, you will have to use a custom port when launching your web application.

Note: Should you have any issues, please stop by the FCS Help Desk located on the main level of the Goldberg Computer Science building.

- In addition to the submission instructions detailed above, there are a few other guidelines you should follow for this assignment:

- You ***must*** use HTML5 semantic document divisions (discussed in class) where possible, instead of simply using divisions .

- You ***must not*** copy / paste code from any websites – this amounts to plagiarism. Do not copy / paste text and content from the websites either.

Note: *In the case you find a piece of code that would be useful for a programming assignment, you may be able to use it if you meet the following requirements.*

Your tutorial/assignment/project must include a ***README.txt*** *file that specifies the following:*

- *The* ***function and line(s) of code*** *(as noted in a Source Code Editor) that include any content taken from a web source.*
- *The* ***web source*** *(i.e., URL) where the code was taken from and the date on which it was accessed.*
- *A brief* ***explanation*** *of what the code is meant to do in its original form (i.e., as it is shown on the web source),*
- *An* ***explanation of how*** *the original* ***code was modified*** *in order to be used in your tutorial/assignment/project.* ***You must have extensively customized the code in order to be able to use it, copy/paste or simply re-naming variables will not suffice.***

- ***Images.*** If you want to use other images on your website, be sure to use images that are published under Creative Commons licenses, i.e. you can use them with proper attribution. A good place to search for such images is on the Creative Commons website: **http://search.creativecommon-**

s.org/ *Always remember to attribute credit to the image creator. Credit should either be in HTML comments or in a separate document named* ***"README.txt"***

- The emphasis in Assignment 3 is for you to apply your knowledge of front-end and back-end web development for creating usable applications where the back-end successfully supports the front-end with which the user interacts.

- You are welcome to include additional features in **A3** such as those that can be achieved through the use of CSS and Javascript. **However, bear in mind the following:**

  - Your submission **must** meet the criteria specified in A3, first and foremost. Beyond this requirement, you are welcome to include additional aspects of future assignments. However, **no bonus points will be granted or replacement will be allowed** for any missing aspects of **A3.**

  - You stand to lose points if the additional markup / CSS elements that you might implement interferes with the basic requirements of **A3.**

  - I will not stop you from exploring beyond what is taught in class or what is expected in these assignments. However, please be mindful of what you submit as your assignment submission.

Academic Integrity[1]

At Dalhousie University, we respect the values of academic integrity: honesty, trust, fairness, responsibility and respect. As a student, adherence to the values of academic integrity and related policies is a requirement of being part of the academic community at Dalhousie University.

What does academic integrity mean?

Academic integrity means being honest in the fulfillment of your academic responsibilities thus establishing mutual trust. Fairness is essential to the interactions of the academic community and is achieved through respect for the opinions and ideas of others. Violations of intellectual honesty are offensive to the entire academic community, not just to the individual faculty member and students in whose class an offence occurs. (See Intellectual Honesty section of University Calendar)

How can you achieve academic integrity?

- Make sure you understand Dalhousie's policies on academic integrity.
- Give appropriate credit to the sources used in your assignment such as written or oral work, computer codes/programs, artistic or architectural works, scientific projects, performances, web page designs, graphical representations, diagrams, videos, and images. Use RefWorks to keep track of your research and edit and format bibliographies in the citation style required by the instructor (See http://www.library.dal.ca/How/RefWorks).
- Do not download the work of another from the Internet and submit it as your own.
- Do not submit work that has been completed through collaboration or previously submitted for another assignment without permission from your instructor.
- Do not write an examination or test for someone else.
- Do not falsify data or lab results.

These examples should be considered only as a guide and not an exhaustive list.

What will happen if an allegation of an academic offence is made against you?

I am required to report a suspected offence. The full process is outlined in the Discipline flow chart, which can be found at: http://academicintegrity.dal.ca/Files/AcademicDisciplineProcess.pdf and includes the following:

1. Each Faculty has an Academic Integrity Officer (AIO) who receives allegations from instructors.
2. The AIO decides whether to proceed with the allegation and you will be notified of the process.
3. If the case proceeds, you will receive an INC (incomplete) grade until the matter is resolved.

[1] Based on the sample statement provided at http://academicintegrity.dal.ca.

4. If you are found guilty of an academic offence, a penalty will be assigned ranging from a warning to a suspension or expulsion from the University and can include a notation on your transcript, failure of the assignment or failure of the course. All penalties are academic in nature.

Where can you turn for help?

- If you are ever unsure about ANYTHING, contact myself.
- The Academic Integrity website (http://academicintegrity.dal.ca) has links to policies, definitions, online tutorials, tips on citing and paraphrasing.
- The Writing Centre provides assistance with proofreading, writing styles, citations.
- Dalhousie Libraries have workshops, online tutorials, citation guides, Assignment Calculator, RefWorks, etc.
- The Dalhousie Student Advocacy Service assists students with academic appeals and student discipline procedures.
- The Senate Office provides links to a list of Academic Integrity Officers, discipline flow chart, and Senate Discipline Committee.