MANITOBA INSTITUTE OF
TRADES AND TECHNOLOGY

Tools and Automation

Course Outline

Course Overview

Course Code: SD-130

Program: Software Developer

Credits: Full Credit

Course Hours: 44 hours

Prerequisites: SD-120

Academic Year: 2018-2019

Class Times & Information

Location: 204 Henlow

Days: Monday to Friday

Times: 8:45 – 1:45 (4 hours/d)

Format: Classroom

Start Date: November 26th, 2018

End Date: December 10th, 2018

Instructor Overview

Instructor: Johnathan Niziol

Email: Johnathan.niziol@mitt.ca

Course Description

Navigating around a computer using a shell is a necessary skill for all software developers and keeping source code safe is just as important. In this course students will learn the basics on using the shell and securely managing their source code using Git, a popular source control application. Finally, students will leverage their new found shell knowledge and build their own tools and scripts to help optimize their development workflow.

Features Udacity Content.

Methods

The methods for instruction will vary, and include:

- Group lecture and/or instruction,
- Demonstrations,
- Research and individual study,
- In-class practice and review,
- Peer support

Materials

- Personal Udacity Account (to be supplied by MITT)
- Laptop Computer (to be supplied by MITT)

General Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Navigate comfortably around an operating system using shell commands.
2. Use basic shell commands to execute applications and scripts.
3. Understand the importance of a Version Control System.
4. Manage source code using a standard Git workflow.
5. Create build scripts and automation scripts that help to increase developer productivity.

MITT Accessibility Statement

MITT is committed to creating a learning environment that meets the needs of its diverse student body. If you anticipate or experience any barriers to learning in this course, please feel welcome to discuss your concerns with me.

If you have a disability, or think you may have a disability, it is strongly recommended for you to meet with Kayla Hoskins, Student Advisor - Accessibility, to begin this conversation or request an official accommodation. You can find more information about the Accessibility Services, including contact information, here: http://www.mitt.ca/student-success/accessibility-services. If you have already arranged accommodations through Kayla Hoskins, please feel free to meet with me if you have any questions or concerns regarding the implementation of your accommodations. If you do not have a documented disability, remember that other support services, including the Learning Specialist, peer tutors and clinical services are available to all students through MITT Student Services.

Course Schedule

Please note that instructors reserve the right to adjust the course schedule without prior notification to meet the changing needs of the class as a whole. It is the responsibility of the student to follow up in cases of missed classes.

Week/Class	Topic(s) with chapters (material) covered	Activities/Assessment
Class 1	**The Shell** ● Installing a Shell ● Basic Shell Commands ● Man pages **File System** ● Hierarchy	Lecture In-Class Exercises **Udacity Content** Lesson 2.9: Shell Workshop
Class 2	**The Shell** ● Advanced Shell Commands	Review Lecture In-Class Exercises Research **Udacity Content**

		Lesson 2.9: Shell Workshop
Class 3	**Git and Github** ● What is version control? ● Git Setup ● Creating a Repo ● Cloning a Repo	Review Lecture In-Class Exercises **<u>Udacity Content</u>** Lesson 2.10: What is Version Control? Lesson 2.11: Creating a Repo **Quiz #1**
Class 4	**Git and Github** ● Adding Changes ● Staging Changes ● Checking repo status ● Committing Changes	Review Lecture In-Class Exercises Research **<u>Udacity Content</u>** Lesson 2.12: Review a Repos History Lesson 2.13: Add Commits to a Repo **Assignment #1**
Class 5	**Git and Github** ● Reviewing a Repo's history ● Tagging, Merging ● Branching Model	Review Lecture In-Class Exercises Research **<u>Udacity Content</u>** Lesson 2.14: Tagging, Branching, Merging
Class 6	**Git and Github** ● Tagging, Merging ● Branching Model ● Undoing Changes ● Managing remote Repo's	Review Lecture In-Class Exercises Research **<u>Udacity Content</u>** Lesson 2.14: Tagging, Branching, Merging

		Lesson 2.15: Undoing Changes **Quiz #2**
Class 7	**Git and Github** ● Working with Github ● Pull requests ● Rebasing	Review Lecture In-Class Exercises Research **<u>Udacity Content</u>** Lesson 2.16: Working With Remotes Lesson 2.17: Working on another Dev's Repo Lesson 2.18: Staying in Sync with A Remote Repo. **Assignment #2**
Class 8	**Build Tools** ● Introduction to Build Tools ● Gulp ● Grunt	**<u>Udacity Content</u>** Lesson 3.14: Introduction to Web Tooling and Automation Lesson 3.16: Powerful Builds **Quiz #3**
Class 9	**Automation** ● Linting ● CSS Minification ● JS Minification ● JS Concatenation	**<u>Udacity Content</u>** Lesson 3.19: Awesome Optimizations
Class 10	**Live Editing** ● Watching files for changes	**<u>Udacity Content</u>** Lesson 3.18: Professional Developer-Fu **Assignment #3**
Class 11	**Review**	**<u>Udacity Content</u>** Lesson 3.17: Expressive Live Editing **Final**

Student Evaluation

Type of Evaluation	Percentage of Grade	Date Due/Assessed
Quizzes	**30%**	
Quiz #1	10%	Class 3
Quiz #2	10%	Class 6
Quiz #3	10%	Class 8
Assignments	**40%**	
Assignment #1	10%	Class 4
Assignment #2	15%	Class 7
Assignment #3	15%	Class 10
Final	**30%**	Class 11

Evaluation Details

Quizzes

Quizzes will test students retention and understanding of the material covered in class as well as material seen in exercise. Quizzes are cumulative in nature and can include questions presented since the beginning of the course, but will tend to focus only the most recently discussed material. Quizzes are all of equal value.

Assignments

Assignments will encompass a combination of theory and practical work, allowing students to apply and demonstrate their newly-acquired knowledge and skills. Unless specified, assignments will be individual efforts.

Assignments for the class will be based on Udacity, exercises and other things discussed in class. Your instructor will clearly highlight which exercises must be submitted for evaluation. Students will receive rubrics for each assignment so they can familiarize themselves on the evaluation criteria.

Students may be provided class time for each assignment, however anything not completed during class time must be completed for homework. Only assignments submitted before the deadline will be evaluated. Assignment solutions will be reviewed as a group immediately after the deadline.

Final Project

The final project is assigned through Udacity and will be clearly indicated. The final project is an individual effort, **no group work is allowed**. Refer to the **Academic Honesty** policy for more information on individual expectations.

The final project in this course will require students to demonstrate their mastery of Git, the shell, and JavaScript build scripts. The final project is cumulative in nature, using all of the material learned throughout the course to construct a project that includes automated test scripts, build scripts, CSS and JS minification, is properly linted and has been saved to the students personal Github account.

In order to obtain course marks, the project must be submitted to the instructor on the due date indicated in the provided rubric. For each 24-hours an assignment is late, students will lose 10% of their final project grade.

Grading

Letter Grade	Grade Point Value	Accumulated Evaluation Percentage
A+	4.5	90 – 100%
A	4.0	80 – 89%
B+	3.5	75 – 79%
B	3.0	70 – 74%
C+	2.5	65 – 69%
C	2.0	60 – 64%
D	1.0	50 – 59%
F	0.0	0 – 49%

Note: A passing grade in this course is 50%.

Course Specific Policies

Laptop Policy

Students are expected to keep their laptop in good running order and are required to bring it to every class. Coming to class without a functioning laptop may require the student to go get the laptop, an absence, and/or affect their participation and professionalism grade.

Academic Honesty

Assignments are intended to be your own work unless explicitly indicated your instructor.

When completing these individual assignments, it is acceptable, and you are **encouraged** to:

- Discuss, at a high level, the problem and potential solution with your fellow students. "At a high level" refers to general approaches to solutions, and not the actual wording of a specific solution.

- Seek assistance in solving small syntax errors from your fellow students, or your instructor.
- Research and obtain solution ideas from the Internet or other external sources.
- Include external code or background material in your solution provided that it has been supplied by the instructor explicitly for use in your solution.

However, it is **not** academically honest to:

- Copy, in whole or in part, any portion of a solution from another student. This is considered plagiarism (unless the instructor explicitly permits group work).
- Copy, in whole or in part, any portion of a solution from the Internet or any other external sources. Even with proper citation, it is generally not acceptable for more than a small percentage of an assignment to be copied verbatim from other sources. The exact threshold is at the discretion of the instructor and MITT.
- Knowingly provide a written solution to an assignment to another student.
- Alter an assignment after it has been graded.

In summary, assignments cannot be copied verbatim from any source. Ideas may be discussed and researched, but ideas must be expressed in your own words, and the code should be your own, unless the instructor explicitly permits group work.

Project-Based Learning Environment

The Software Development industry is largely based on completing projects within a variety of timelines in both teams and independently. Students will prepare themselves for industry by practicing the skills required for this type or workplace including; time-management, problem-solving, and showing adaptability by working with various deadlines and team members. The instructor will act as a facilitator to guide students through the content (both Udacity and other) but students will be responsible for ensuring that they are keeping up with the course schedule and learning outcomes by submitting quality assignments and projects on time.

MITT Academic Policy and Regulation

Students are responsible for reviewing and observing all MITT Student Policies while engaged in any form of academic activity with the Institute and should refer to the MITT website for all policy information.

Key policies to refer to in relation to this course include:

- Student Discipline
- Student Behaviour
- Student and MITT Expectations
- Attendance Policy
- Documentation Requirements
- Dress Code Policy

Academic Integrity

As per the MITT Academic Integrity Policy, academic dishonesty in any form is unacceptable. This policy applies to all courses at MITT and defines all activities and behaviours that might constitute grounds for an academic violation.

MITT expects all students to attend an academic orientation session within their program and to adhere to the principles of academic integrity.

Students found to be in violation of the Academic Integrity Policy will be subject to disciplinary action as defined by the MITT Student Discipline Policy. Refer to both of these policies for further details.

Retention of Course Outline

Students are advised to retain course outlines for future use in support of applications for employment or transfer of credits.

Information contained in this Course Outline is correct at the time of publication. Continuous improvement is important to MITT and our program delivery. Program and course content may be revised on an ongoing basis to ensure relevance to changing educational and/or labour market needs. As such, this program may be subject to change and the information outlined within should not be viewed as a representation or guarantee of offering.