

**Course Outline**   **Faculty:**

**Anton Kovunov**

[Anton.kovunov@humber.ca](mailto:Anton.kovunov@humber.ca)

Course Name: Web Application Development for CPAN Diploma **Associate Dean:**

CPAN 228 Jonathan Kim

jonathan.kim@humber.ca

**Academic Year**: 2023-2024

**Schedule Type Code:** LLB

# Land Acknowledgement

Humber College is located within the traditional and treaty lands of the Mississauga of the Credit. Known as Adoobiigok, the “Place of the Black Alders” in Michi Saagiig language, the region is uniquely situated along the Humber River Watershed, which historically provided an integral connection for Anishinaabe, Haudenosaunee, and Wendat peoples between the Ontario Lakeshore and Lake Simcoe/Georgian Bay regions. Now home to people of numerous nations, Adoobiigok continues to provide a vital source of interconnection for all.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **School** | Faculty of Applied Sciences & Technology | | | |
| **Program** | Computer Programming (CPAN) | | | |
| **Course Name:** | Web Application Development for CPAN Diploma (CPAN 228) | | | |
| **Pre-Requisite(s)** | CPAN 228-Data Structures and Algorithms | | | |
| **Co-Requisite(s)** | none | | | |
| **Pre-Requisite(s) for** |  | | | |
| **Equates** | none | | | |
| **Restrictions** | Post-Graduate; Faculty of Applied Sciences and Technology | | | |
| **Credit Value** | 4 | | | |
| **Total Course Hours** | 96 | | | |
| **Developed By:** | **Prepared By:** | **Approved by** |
| Anton Kovunov | Anton Kovunov | Jonathan Kim |

# Course Description

Advanced Java course that involves using Spring Boot as a high-level framework, working on building web applications, creating database integration using ORM and working with repositories of different kind and granularity. Touching on the topic of security, REST API and configuration that helps to increase overall understanding on how to build enterprise level applications and how to do it fast.

# Course Rationale

Building applications and delivering them to production is not a trivial task, but fortunately, we have frameworks that encapsulate most of the heavy lifting and provide us with the tools to describe applications on the highest level possible with various levels of instrumentation and granularity. Knowledge of tools available for us in Spring Boot helps to build the sturdy foundation for students to write applications that can be used in production.

# Program Outcomes Emphasized in this Course for CP221

* Contribute to the development, documentation, implementation, maintenance and testing of software systems by using industry standard software development methodologies based on defined specifications and existing technologies/frameworks.

**Program Learning Outcomes Emphasized in this Course for CP311:**

* Analyze, develop and maintain robust computing system solutions through validation testing and industry best practices

# Course Learning Method(s)

Action Learning

Problem-Based Learning (PBL) Lecture

# Learning Outcomes and Assessments

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Outcome** | **Lesson Objectives** | **Summative Assessments** | **Formative**  **Assessments** |
| Being able to create applications skeleton using Spring Initializr with Maven as a build tool | Create application using Spring Initializr, know building blocks that create app, being able to run it | Performance/Exhibition/Demonstration:  Create application using Spring Initializr, know building blocks that create app, being able to run it using maven | Lab: create an app with sample controller and run it with maven, confirm program correctness |
| Being able to create a simple controller using Spring Web and Thymeleaf | Add controller to our app skeleton and connect it to the template | Performance/Exhibition/Demonstration:  Use Thymeleaf engine to create a web page, create a Spring Component that acts as a controller and display the page | Lab: Display message on the web page that has correct URL and correct styling |
| Being able to operate with Model Attributes, attach them to the template and do manipulations with them using HTML form with Thymeleaf flavour, validate form input | Add model to the template, fill in the form and execute request to see that fields are populated and passed correctly | Performance/Exhibition/Demonstration:  Add POJO model to the template, connect template to controller, fill in the form and execute request to see that fields are populated and passed correctly | Lab:  Make sure on form submission model is sent to the controller and populate list by using session attributes  Assignment: Create project using Spring Web and validate input and form submission |

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Outcome** | **Lesson Objectives** | **Summative Assessments** | **Formative**  **Assessments** |
| Being able to connect JDBC to Spring Boot project, being able to execute simple queries | Add JDBC capability to Spring Boot project, execute simple queries | Performance/Exhibition/Demonstration: Connect H2 database to project, use JDBC library and create simple queries using JDBCTemplate. | Written Assessment:  Connect in class project form submission to execute database queries |
| Being able to use Spring Data JDBC, debug database queries using H2 Console and operate with CrudRepository abstraction | Use Spring Data Jdbc library, streamline query creation and add more granular search | Performance/Exhibition/Demonstration:  Switch dependency to Spring Data JDBC, explain benefits of using more concise approach and replace previous queries | Lab:  Add extra functionality to the project with custom query |
| Being able to use Spring Data JPA, add custom queries, use sorting and pagination | Replace Spring Data JDBC with JPA, change models to correspond to JPA specification | Performance/Exhibition/Demonstration:  Create repository using CRUD template and a separate one using Pagination and Sorting, demonstrate usage and pros/cons | Lab: Create a list of entities and display them on a page, add pagination.  Assignment: Add database capabilities to personal project, extra queries |
| Being able to add spring security to the project, configure simple security rules | Add Spring Security dependency, provide bean that controls security configuration | Performance/Exhibition/Demonstration:  Configure Spring Security, add User model to store username and password, create simple configuration with CSRF exceptions | Lab:  Add simple registration form, make sure you can only see pages when registered |
| Being able to customize security with custom made login pages | Create additional config with redirects and custom login controller and page | Performance/Exhibition/Demonstration:  Create security config adjustments that have login and redirects additions | Lab:  Wire login and redirect to custom login page and make sure pages have granular role protection  Assignment:  Add multiple roles to the assignment project, configure management page for administrator |
| **Learning Outcome** | **Lesson Objectives** | **Summative Assessments** | **Formative**  **Assessments** |
| Being able to create simple rest controllers, perform HTTP requests towards them | Create simple controller with REST capabilities | Performance/Exhibition/Demonstration:  Create REST Controller with GET, POST, PUT and DELETE methods, being able to execute them using curl or POSTMAN | Lab: Create REST Controller with GET, POST, PUT and DELETE methods, being able to execute them using curl or POSTMAN for the lab project |
| Being able to consume other rest api, familiarize with Microservices pattern, use configuration and Spring Profile | Have two REST services communication with each other | Performance/Exhibition/Demonstration:  Create new project with plain REST functionality, consume the API from third-party vendor | Lab: Create a dashboard project that exposes REST endpoints, consume the data from it and format it per requirements |

**Assessment Summary Weighting**

| **Summary** | **Weight** |
| --- | --- |
| Total for all Lab Exercises (Total 10) | 40% |
| Total for all Assignments (Total 3) | 30% |
| Final Project | 30% |
| **Total** | 100% |

# Modules of Study

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** | **Course Learning Outcomes** | **Resources** | **Assessments** |
| Introduction to  Spring Boot and Maven | Being able to create applications skeleton using Spring Initializr with Maven as a build tool | Presentation, Recording | Performance/Exhibition/Demonstration:  Create application using Spring Initializr, know building blocks that create app, being able to run it using maven  Lab: create an app with sample controller and run it with maven, confirm program correctness |
| Spring Web Basic | Being able to create a simple controller using Spring Web and Thymeleaf | Presentation, Recording | Performance/Exhibition/Demonstration:  Use Thymeleaf engine to create a web page, create a Spring Component that acts as a controller and display the page.    Lab: Display message on the web page that has correct URL and correct styling |

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** | **Course Learning Outcomes** | **Resources** | **Assessments** |
| Spring Web Advanced | Being able to operate with Model Attributes, attach them to the template and do manipulations with them using HTML form with Thymeleaf flavour, validate form input | Presentation, Recording | Performance/Exhibition/Demonstration:  Add POJO model to the template, connect template to controller, fill in the form and execute request to see that fields are populated and passed correctly  Lab:  Make sure on form submission model is sent to the controller and populate list by using session attributes  Assignment: Create project using Spring Web and validate input and form submission |
| Spring JDBC | Being able to connect JDBC to Spring Boot project, being able to execute simple queries | Presentation, Recording | Performance/Exhibition/Demonstration: Connect H2 database to project, use JDBC library and create simple queries using JDBCTemplate.  Lab:  Connect in class project form submission to execute database queries |
| Spring Data JDBC | Being able to use Spring Data JDBC, debug database queries using H2 Console and operate with CrudRepository abstraction | Presentation, Recording | Performance/Exhibition/Demonstration:  Switch dependency to Spring Data JDBC, explain benefits of using more concise approach and replace previous queries  Lab:  Add extra functionality to the project with custom query |
| Spring Data JPA | Being able to use Spring Data JPA, add custom queries, use sorting and pagination | Presentation, Recording | Performance/Exhibition/Demonstration:  Create repository using CRUD template and a separate one using Pagination and Sorting, demonstrate usage and pros/cons  Lab: Create a list of entities and display them on a page, add pagination.  Assignment: Add database capabilities to personal project, extra queries |
| Spring Security | Being able to add spring security to the project, configure simple security rules | Presentation, recording | Performance/Exhibition/Demonstration:  Configure Spring Security, add User model to store username and password, create simple configuration with CSRF exceptions  Lab:  Add simple registration form, make sure you can only see pages when registered |
| Spring Security Advanced | Being able to customize security with custom made login pages | Presentation, Recording | Performance/Exhibition/Demonstration:  Create security config adjustments that have login and redirects additions  Lab:  Wire login and redirect to custom login page and make sure pages have granular role protection  Assignment:  Add multiple roles to the assignment project, configure management page for administrator |
| **Module** | **Course Learning Outcomes** | **Resources** | **Assessments** |
| Spring REST | Being able to create simple rest controllers, perform HTTP requests towards them | Presentation, Recording | Performance/Exhibition/Demonstration:  Create REST Controller with GET, POST, PUT and DELETE methods, being able to execute them using curl or POSTMAN  Lab: Create REST Controller with GET, POST, PUT and DELETE methods, being able to execute them using curl or POSTMAN for the lab project |
| Spring REST Template | Have two REST services communication with each other, understand how this communication occurs | Presentation, Recording | Performance/Exhibition/Demonstration:  Create new project with plain REST functionality, consume the API from third-party vendor  Lab: Create a dashboard project that exposes REST endpoints, consume the data from it and format it per requirements |
| Spring Config | Add configuration, being able to use environment abstraction | Presentation, Recording | Performance/Exhibition/Demonstration:  Use profiles to conditionally use Spring Components, understand environment abstraction  Lab: Spin up Docker Image for Postgres Data Source and make sure H2 and Postgres are used interchangeably when we switch profiles |
| Final Project | Work as a unit in teams of 4, combine all the skills learned in a full-stack project with Security, REST, Web and Database abstractions, configure different profiles and understand usability | Presentation, Recording | Project: Create a project with database usage, Spring Web multiple pages with transitions, consume other REST API to get the data, secure project with roles and login/registration pages, use different data source for different profiles |

**Required Resources**

Presentations, Recordings, additional resource: Spring in Action Chapter 6, available in open source

# Additional Tools and Equipment

Laptop to program specifications.

JDK and Maven (open source tools provided by Oracle and Apache respectively)

Visual Studio Code (IDE by Microsoft as free download)

# Essential Skills

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Skills** | **Measurement** | **Details** |
| Communication | Reading  Writing  Listening | Reinforce and measure | Journal entries will be provided for each lab. Students need to reinforce what they did in each lab.  Submitted for each lab to reinforce the labs objectives. |
| Numeracy | Understanding and applying mathematical concepts and reasoning  \_ | Teach and measure | Certain aspects of Spring Boot operations apply mathematics to obtain results. The student will need to know how to use calculations to obtain the correct results.  Problems given to students for solutions to be coded can contain mathematical statements. |
| **Section** | **Skills** | **Measurement** | **Details** |
| Critical  Thinking and  Problem-  Solving | Analysing  Evaluating  Decision-Making | Teach and measure | Since this is programming language students are writing code to resolve a problem. This involves the skills identified to accomplish this.  The code students submit as a resolution for a given problem are graded for accuracy and efficiency. |
| Information  Management | Gathering and managing information Selecting and using appropriate tools and technology for a task or project  Computer  literacy | Teach and measure | Since this is an IT course students need to use appropriate code in their Spring Boot program to resolve the issue. This can involve using the correct technology in the syntax.  Through labs and assignments this is being assessed. |
| Interpersonal  Skills | Networking | Reinforce and measure | Students are encouraged to discuss problems and arrive at their own solutions to problems given in class. In a lab setting students are being observed as they work.  Students can be observed as they work. |
| Personal Skills | Managing change and  being flexible and adaptable Engaging in reflective practice  Demonstrating personal responsibility | Reinforce and measure | Students need to retain concepts learned. In programming the concepts presented do not go away after they have been presented. Students need to show responsibility to retain the facts of what they are learning. They need to be willing to investigate different paths to solving problems. Through direct observation, and through assignments and tests given you can see whether students are retaining the material. |

**Prior Learning Assessment Recognition (PLAR)**

PLAR is not available for this course.

# Academic Regulations

It is the student's responsibility to be aware of the College Academic Regulations. The Academic Regulations apply to all applicants to Humber and all current students enrolled in any program or course offered by Humber, in any location.

Information about academic appeals is found in the Academic Regulations.

# Accessible Learning Services

Humber strives to create a welcoming environment for all students where equity, diversity and inclusion are paramount. Accessible Learning Services facilitates equal access for students with disabilities by coordinating academic accommodations and services. Staff in Accessible Learning Services are available by appointment to assess specific needs, provide referrals and arrange appropriate accommodations. If you require academic accommodations, contact:

Accessible Learning Services: <http://www.humber.ca/student-life/swac/accessible-learning>

North Campus: (416) 675-6622 X5090

Lakeshore Campus: (416) 675-6622 X3331

# Academic Integrity

Academic integrity is essentially honesty in all academic endeavors. Academic integrity requires that students avoid all forms of academic misconduct or dishonesty, including plagiarism, cheating on tests or exams, or any misrepresentation of academic accomplishment.

# Disclaimer

While every effort is made by the professor/faculty to cover all material listed in the outline, the order, content, and/or evaluation may change in the event of special circumstances (e.g. time constraints due to inclement weather, sickness, college closure, technology/equipment problems or changes, etc.). In any such case, students will be given appropriate notification in writing, with approval from the Dean (or designate) of the School.

# Copyright

Copyright is the exclusive legal right given to a creator to reproduce, publish, sell or distribute his/her work. All members of the Humber community are required to comply with Canadian copyright law which governs the reproduction, use, and distribution of copyrighted materials. This means that the copying, use, and distribution of copyright-protected materials, regardless of format, is subject to certain limits and restrictions. For example, photocopying or scanning an entire textbook is not allowed, nor is distributing a scanned book.

See the Humber Libraries website ([http://library.humber.ca)](http://library.humber.ca/) for additional information regarding copyright and for details on allowable limits.

Humber College Institute of Technology and Advanced Learning • 2019/2020 This document is available in an alternate format upon request.