

TEC-060: Back End Software Developer

Professional Education Course Syllabus

Course Description

Back End Software Developers are in charge of building applications that do the behind-the-scenes work. This typically involves receiving requests from a Front End application, processing the request, reading and writing to a database, and responding back to requests from other applications.

In this 18-week, Back End Software Development program, students take three courses to learn the skills and technologies necessary to enter the industry as a Back End Software Developer: Introduction to Java, Relational Databases with MySQL, and Web API Design with Spring Boot. Skills and technologies learned will include Java 1.8, Algorithms, OOP, Design Patterns, Unit Testing, SQL, DDL, DML, Database Concepts, JDBC, Spring Boot, Spring Data, REST, JPA, JWT, AWS, and more.

Program Objectives

Week	Objectives
1 Intro to Java	Students will: <ul style="list-style-type: none"> Learn to navigate using the CLI Understand and implement source control principles and workflows using Git and GitHub Understand, at a high level, what programming is Learn about Java and its place in the technology world Learn about variables and data types Learn common variable operations
2	Students will: <ul style="list-style-type: none"> Learn Boolean logic Learn how to use Boolean logic in Java conditional statements Learn how and when to utilize different loops Learn how to read and respond to user console input
3	Students will: <ul style="list-style-type: none"> Learn how and when to use arrays Learn how to modularize code using methods Learn how to use preexisting Java Objects and their methods and properties Learn how to read and understand Java documentation

	<ul style="list-style-type: none"> Learn the difference between the equality operator and the .equals() method
4	<p>Students will:</p> <ul style="list-style-type: none"> Study the differences between Strings and StringBuilders and learn when to use which; Learn about generics; Learn about collections including how and when to use lists, maps and sets, as well as the differences between them.
5	<p>Students will:</p> <ul style="list-style-type: none"> Study Object Oriented Programming (OOP) principles; Learn how and when to create classes; Learn how and when to utilize class inheritance; Understand Java access modifiers; Learn how and when to use interfaces; Understand exceptions and how to handle them.
6	<p>Students will:</p> <ul style="list-style-type: none"> Learn how to use the debugger and other debugging skills Learn about unit tests, Test Driven Development (TDD), and how to write unit tests
7 Relational Databases with MySQL	<p>Students will:</p> <ul style="list-style-type: none"> Learn about relational databases and their place in the web application architecture Learn database CRUD operations and other commands including SELECT, INSERT INTO, UPDATE, DELETE, WHERE, LIKE, IN, and ORDER BY
8	<p>Students will:</p> <ul style="list-style-type: none"> Learn about table relationships and foreign keys Learn how to query the database using joins Explore advanced queries using functions and the GROUP BY and HAVING clauses
9	<p>Students will:</p> <ul style="list-style-type: none"> Study database design concepts Learn how to create a database Learn how to create and drop tables Learn when and how to properly normalize a database schema
10	<p>Students will:</p> <ul style="list-style-type: none"> Learn about additional database constraints Learn when and how to create views Learn how to utilize indexes to improve query performance Learn how to create and use stored procedures
11	<p>Students will:</p> <ul style="list-style-type: none"> Learn how to connect to a database from an application Build an application which reads and writes to a database

12	Students will: <ul style="list-style-type: none"> • Work in a team setting to build a full CLI CRUD application
13 Capstone Course: Web API Design with Spring Boot	Students will: <ul style="list-style-type: none"> • Learn RESTful concepts • Learn about HTTP and its verb methods • Learn how to use Postman to send HTTP requests • Design API documentation for final project • Improve existing skills by working on the Final Project
14	Students will: <ul style="list-style-type: none"> • Install and use maven to build applications and manage dependencies • Learn how to set up a Spring Boot application • Build API endpoints (controller layer only) using Spring Boot • Improve existing skills by working on the Final Project
15	Students will: <ul style="list-style-type: none"> • Learn how to expand their API endpoints by adding Service and Data Access layers to their applications • Learn how to set up and use JPA • Build full CRUD applications • Improve existing skills by working on the Final Project
16	Students will: <ul style="list-style-type: none"> • Learn how to enable, configure, and use application logs • Improve existing skills by working on the Final Project
17	Students will: <ul style="list-style-type: none"> • Understand basic security concepts such as salting and hashing • Study JSON Web Tokens (JWT) • Improve existing skills by working on the Final Project
18	Students will: <ul style="list-style-type: none"> • Learn about Amazon Web Services (AWS) console and features • Deploy applications to AWS • Improve existing skills by working on the Final Project • Prepare and present their Final Project

Grading Breakdown

Below is a breakdown of the assignments and quizzes that make up your final grade. The minimum grade required to successfully pass the program is a C or 70%, students must also complete and turn in each assignment to pass.

Week	Assignment	Points	% of Total Grade
1	Research Topics	30	1
1	Coding Project	70	2.5
2	Research Topics	30	1
2	Coding Project	70	2.5

3	Research Topics	30	1
3	Coding Project	70	2.5
3	Quiz	100	3.5
4	Research Topics	30	1
4	Coding Project	70	2.5
5	Research Topics	30	1
5	Coding Project	70	2.5
6	Research Topics	30	1
6	Coding Project	70	2.5
6	Quiz	100	3.5
7	Research Topics	30	1
7	Coding Project	70	2.5
8	Research Topics	30	1
8	Coding Project	70	2.5
9	Research Topics	30	1
9	Coding Project	70	2.5
9	Quiz	100	3.5
10	Research Topics	30	1
10	Coding Project	70	2.5
11	Research Topics	30	1
11	Coding Project	70	2.5
12	Research Topics	30	1
12	Coding Project	70	2.5
12	Quiz	100	3.5
13	Research Topics	30	1
13	Coding Project	70	2.5
14	Research Topics	30	1
14	Coding Project	70	2.5
15	Research Topics	30	1
15	Coding Project	70	2.5
15	Quiz	100	3.5
16	Final Project Work	100	3.5
16	Scrum Retro	50	1.8
16	Root Cause Analysis	50	1.8
17	Final Project Work	100	3.5
17	Scrum Retro	50	1.8
17	Root Cause Analysis	50	1.8
18	Final Project Work	100	3.5
18	Project Retro	50	1.8
18	Demo	300	10.5

Research Topics

Research Topics are an informal essay that requires the student to research the topics presented in the lesson and find resources to deepen their understanding of said topics. The technology industry is based on constantly searching for information; these assignments are developed to help students acquire the skills necessary to be successful as a developer in this aspect. Research Topics account for 15% of your total grade.

Rubric

Category	Criteria	% of Grade
Accuracy	Is the information accurate?	25
Organization	Is the essay clean and organized? Ideas are presented in a logical order.	25
Citations	Students reference and cite at least 5 sources.	25
Completeness	All requirements of the assignment are complete.	25

Coding Projects

The weekly Coding Projects are a vital component of the program and are worth an accumulative 37.5% of your overall grade. These assignments afford students hands on experience with the subject matter and help students develop the skills necessary to become proficient developers.

Rubric

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Quizzes

The quizzes within this program are designed to demonstrate mastery of the topics covered. The quizzes are open book, internet, and all other resources *except* other students. Questions on the quizzes are to be answered solely by the student. While the quizzes are open book, plagiarism is not acceptable. If a student finds an answer to a question, he/she must express the answer in his/her own words or code. Do not copy and paste. Quizzes account for 17.5% of your total grade.

Final Project

The Final Project is the culminating activity of the program. It is used to demonstrate mastery of the learned materials, as well as the ability to apply the knowledge and skills learned throughout the program. The Final Project, along with all its components, is worth 30% of your final grade. The Final Project spans over the last 3 weeks of the program. This project may be a group project or individual project based on your instructor's discretion. The instructor may assign you a project or allow you to choose your own subject to instructor approval. The Final Project is broken up into 5 distinct categories – Final Project Work, Scrum Retro, Root Cause Analysis, Project Retro, and the Demo.

Final Project Work

The Final Project Work is the actual coding of the project. It is graded as follows:

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete that were committed to for the given week.	25

Scrum Retro

Scrum Retro (retrospective) is a written assignment that allows the student to use Agile tools and processes to analyze the previous week and identify areas for improvement as well as areas of accomplishment.

Category	Criteria	% of Grade
Areas for Improvement	Student identified true areas for improvement as well as plans to make improvements a reality.	25
Areas of Accomplishment	Student accurately identifies areas that he/she succeeded in and outlines plans to continue the success.	25
Organization	Thoughts are concise and clear.	25
Critical Thinking	Student shows deep thought in the expressed ideas.	25

Root Cause Analysis

The Root Cause Analysis is a written assignment that requires students to identify difficulties encountered during the project phase and use critical thinking and research to identify the root cause of the difficulties, as well as the resolutions to the problems.

Category	Criteria	% of Grade
Critical Thinking	Student identifies difficulties and blockers met during the week's project and shows an understanding of why they occurred. Student exhibits problem solving skills by following the problem to the root cause and identifying solutions to overcome the problem.	34
Citations	Student references sources used to overcome the issues outlined.	33
Organization	Thoughts are concise and clear.	33

Project Retro

The Project Retro is similar to the Scrum Retro, but focusses on the entire project and the student's goals moving forward.

Category	Criteria	% of Grade
Areas for Improvement	Student identified true areas for improvement as well as plans to make improvements a reality.	25
Areas of Accomplishment	Student accurately identifies areas that he/she succeeded in and outlines plans to continue the success.	25
Organization	Thoughts are concise and clear.	25
Critical Thinking	Student shows deep thought in the expressed ideas.	25

Demo

The Demo is the pinnacle of the program. It provides the student an opportunity to present the culmination of everything learned and applied throughout the program and the Final Project phase. The Demo will be a live presentation to an audience, including the professor, and will be graded based on the following criteria.

Category	Criteria	% of Grade
Presentation	Student presents well through tone, annunciation, and professional mannerisms.	20
Engagement	The presentation is exciting and engaging. Student avoids monotone speaking. Visuals are utilized and show creativity.	20
Organization	Presentation is concise and clear; the progression makes sense and is easy to follow.	20
Functionality	Completed features work as intended.	20
Completeness	All main features are complete and present in the demo.	20

Late Policy

Assignments turned in late will result in a 10% deduction of points per day past the due date. After ten days, no points will be awarded for the assignment. However, every assignment must still be turned in to pass the program.