

ANTIOCH UNIVERSITY LOS ANGELES  
UNDERGRADUATE STUDIES DIVISION

Course Syllabus

<i>Course title:</i>	Introduction to Full Stack Web Development
<i>Course prefix and number:</i>	CSC 1010, section Sab1
<i>Course Pre-requisite:</i>	None
<i>Location:</i>	Antioch's 400 Corporate Pointe campus, Room A1010
<i>Number of units:</i>	12
<i>Quarter and Year:</i>	Spring 2018
<i>Day/Time:</i>	12 Hours/week of instructional time. Tuesday, Thursday, Friday, 7 p.m. – 10 p.m. (Tuesday, April 3 to Friday, June 22)
<i>Faculty name and degree:</i>	Daniel Gonzalez, MA
<i>Faculty availability (office hours):</i>	By appointment
<i>Faculty contact information:</i>	Instructor will provide
<i>Teaching Associate(s):</i>	

**COURSE DESCRIPTION:**

The course will give students the knowledge and comprehension to fully understand the responsibilities of a Full Stack Web Developer. The course will teach students to utilize the basic software applications and tools used by industry professionals to develop, debug and design web applications.

Students will be taught the basic knowledge and skills that enables them to develop into a Full Stack Web Developer; a programmer with a complete technical profile that covers client side, server side-middle tier, and backend-database developments. The course will give students the basic knowledge needed to understand and participate in the main tasks of Web development: designing, and debugging software that runs in a cross-browser environment, on a web server leading into a database server.

After completing the intro course, the student will have a basic knowledge in API, HTML/CSS, C#, JavaScript, JQuery and Database Development on the Web, among others.

The course will teach students how to work in server, network, and hosting environments. They will obtain the knowledge to understand the concepts in relational and non-relational databases and how they interact with API's and the outside world. They will also be introduced to interface and user experiences, quality assurance, security concerns, and understanding customer and client needs.

Specifically, the course will produce a programmer who is prepared to apply the concepts and skills learned to a more complex scenario, understanding many of the most popular design patterns: APIs, libraries and technologies including but not limited to: HTML5, CSS3, JavaScript, Bootstrap, MVVM, MVC, Angularjs, jQuery,

Ajax, Inversion of Control, Principles of Object Oriented Design, Data Structures, Data Access, Database Design and Architecture, GIS, SMS, SMTP and RESTful Api Design.

## ANTIOCH UNIVERSITY LOS ANGELES UNDERGRADUATE LEARNING OBJECTIVES

All of AULA's undergraduate programs infuse curriculum with this purpose and these values through learning activities that cultivate the following intellectual and practical skills, applied learning, social awareness and responsibility:

- Critical and analytical thinking ability
- Ability to understand issues from multiple perspectives
- Ability to connect learning to lived experience
- Social and intercultural awareness
- Civic and community engagement
- Core competency in foundational skills

### *WEB DEVELOPMENT PROGRAM LEARNING OBJECTIVES*

The AULA/Sabio Web Development program consists of a period of twenty-four (24) weeks during which students complete two 12 week intensive courses: "Intro to Full Stack Web Development" and "Immersive Full Stack Web Development". The overall learning objectives of the Web Development program are to develop competent Full Stack Web Developer professionals. After completing this program, students will possess:

- Progressive beginning, intermediate, and advanced knowledge, skills, and abilities to lead the development and integration of state-of-the-art web software and systems.
- Abilities and skills on all tiers (client/server/backend-database) to be able to create and manage fully effective web based applications.
- Knowledge to apply the Software Development Life Cycle to design, develop, implement and maintain the knowledge and skills learned in the proper order to develop functional web systems.
- Understanding of the changes that systems go through due to social and technological phenomenon, and how to cope with these factors.

### INTRODUCTORY COURSE LEARNING OBJECTIVES/OUTCOMES

After completing the introductory 12 week course titled "Introduction to Full Stack Web Development", students will possess the knowledge and skills to participate in the development and integration of software components for Web application development. Students will possess the knowledge and comprehension of the new tools and technologies by working hands-on with the most common Web applications and tools used by the

industries. Students will possess the ability to evaluate the new and existing functionalities of web based systems and determine the best options available.

The main learning objectives/outcomes for the “Introductory Course on Full Stack Web Development” are:

- Fundamental knowledge and skills to participate in the development of the three major Web application tiers: Client side, Server Side (middle tier) and Backend (Database) development.
- Basic to intermediate programming knowledge and skills in C#, Java, SQL, and HTML, among others, to apply to Web development environments.
- Knowledge and use of integrated developer environments (IDE’s) such as Visual Studio; and Database Management Systems such as SQL Server.
- Fundamental knowledge of Database Design and Architecture, Key Table design considerations, Query Structure and Optimization, and Stored Procedures.
- Ability to apply the lessons-learned to participate in the active development of cross-browser applications through the use of tools such as HTML5, CCS3, JavaScript, jQuery, Knockout.js and AngularJs or another client side MVC and/or MVVM application framework
- Knowledge to build a service oriented (SOA) application program interface (API) and the corresponding back end database through use of: ASP.net C#/.Net or similar middle tier language and framework, RESTFUL API Design, MVC, Data Access, Principles of OOD, Singleton, Caching Strategies, among others.
- Knowledge of current leading developer tools and API's such as AWS, SendGrid, Google API's, Git, Trello, Twilio, and Browser Developer Tools.
- Knowledge to apply the skills learned to all platforms, the different options for each of the layers, and how to evaluate them for creating their own projects.
- Understanding how to utilize team communication and development strategies such as Agile/Scrum Methodologies, Technical Project Management, Product Development, and Team & Project Based Learning, Pair Programing and Developer Code Reviews.
- Ability to communicate effectively between team members and to participate actively and responsibly in a team project understanding their role in the project goals.

## EVALUATION CRITERIA

After the Introductory course students should be able to fully understand the duties of a typical Full Stack Web Developer. Students should recognize a full stack web developer as a programmer with a complete technical profile that covers client side, server side-middle tier, and backend-database developments.

Students’ knowledge and understanding of how a web application gets done from a concept, to the design, to the finished product, will be evaluated. They should demonstrate through class work, projects, assignments, and a final interview, the knowledge and understanding to explain all of the layers of the web development environment, the concepts at play within each layer, and how all of them interact. They should recognize the different options for each of the layers, and how to apply them for their own projects.

The students will be evaluated based on their:

- Knowledge and understanding of the development, design and debugging of software that runs in a cross-browser environment on a web server backed by a database server.

- Full understanding of the SDLC from concept to finished product
- Knowledge and skills to actively participate in web based solutions as part of a team, or as an individual.
- Understanding of class work and lectures and the ability to apply the lessons learned to a web development environment.
- Understanding on the development of relational and non-relational databases and how they interact with API's and the outside world.
- Understanding of quality assurance, security concerns, and customer/client needs
- Participation in class and completion of all assignments, practice work, and lectures on the topics.
- Students' tasks will continuously be evaluated throughout the course as Instructor looks for a progressive improvement in style, technique and application of the subject matters.

In summary, work done in class, class participation in key topics, and assigned readings and projects, will be the main criteria used to evaluate students' knowledge and understanding of the multi-tier web environment and the most popular design patterns. This includes APIs, libraries and technologies including but not limited to: HTML5, CSS3, JavaScript, Bootstrap, MVVM, MVC, Angularjs, jQuery, Ajax, Inversion of Control, Principles of Object Oriented Design, Data Structures, Data Access, Database Design and Architecture, GIS, SMS, SMTP and RESTful Api Design.

Students will also participate in a mock interview where they will be presented with questions on many of the topics covered in the course. Questions will include white boarding coding solutions, open end software design questions as well as questions regarding software development principles and practices. Students are expected to perform at a level that they can communicate their introductory knowledge and understanding on the topics and apply the basics that were covered to a development project. Students have to prove to the Instructor that they are ready for the Immersive Course.

## ASSIGNMENTS

There will always be reading and written/projects assignments for each of the main topics covered in each phase. Sabio readings are developed by Sabio Instructors and staff among others.

### **READING ASSIGNMENTS**

Reading assignments can be found as part of the readings and projects the student will access our online learning management system (LMS)

<http://lms.sabio.la/p/csc-1010/>

#### **For Each Course Phase:**

- Reading Assignments will include at least 2-3 articles to be assigned, handed out in class, or posted as links.
- Writing/Project Assignments will include from 2-3 assignments/projects related to the week's topic and will be due with the specific expectations handed out in class.

## REQUIRED TEXT AND READINGS

Students are required to purchase this book as soon as possible:

## JavaScript and JQuery: Interactive Front-End Web Development

by Jon Duckett

Link: <http://amzn.com/1118531647>

Instructor will provide some guidance in reading this book. It is intended to ensure you have a more solid foundation of JavaScript programming.

The following link leads to the material that is detailed below:

<http://lms.sabio.la>

### **Topics and Assignments**

This is an interactive course where topics and lectures are accessed through the Sabio wiki and google doc links that lead to more directly related material on the Web.

### **Initial Lectures**

- Hardware and Software Requirements
- Pre-Work Guide-Read Me

Gives the student a logical order for the readings on certain varied topics covered in the Intro course:

#### **Pework 100**

Introduction to HTML, CSS, JavaScript:

<http://lms.sabio.la/courses/prework-javascript/lectures/2264920>

#### **Pework 200**

Read the JavaScript book discussed at the link below. Specific pages and chapters are called out:

<http://lms.sabio.la/courses/prework-javascript/lectures/2264880>

#### **Pework 300**

Mini Blogging Pages

<http://lms.sabio.la/courses/prework-javascript/lectures/2264864>

Bootstrap

<http://lms.sabio.la/courses/prework-javascript/lectures/2264921>

Pework Extra

Only when you are done with the rest of the work.

<http://lms.sabio.la/courses/prework-javascript/lectures/2264862>

<http://lms.sabio.la/courses/prework-javascript/lectures/2264863>

<http://lms.sabio.la/courses/prework-javascript/lectures/2264938>

### **Pre-Work C# by Level**

Read "Prework Guide for C#".

Visual Studio Installation Video:

The following readings and assignments, projects, and topics listed by level are covered in the course:  
<http://lms.sabio.la/courses/prework-c/lectures/2264949>

- 130 NetLab
- 210 C Sharp Basics
- 230 Static Classes
- 240 Non-Static Classes
- 310 MVC Intro
- 320 MVC Blog Part 1
- 330 MVC Blog Part 2
- 340 MVC Blog Part 3
- 350 - MVC Blog Part 4

***Pre-work-Database***

SQL Server 2016 Express Installation Video:  
<http://lms.sabio.la/courses/enrolled/152806>

Pework 210- W3Schools

Pre-Work Guide for Database  
<http://lms.sabio.la/courses/enrolled/153011>

- 320 Data Modification
- 340 Stored Procedures

## **Pre-Work Java-Script**

Required book (Available from [Antioch's online bookstore](#) and other sources):

Duckett, Jon. *JavaScript and JQuery: Interactive Front-End Web Development*. Hoboken, NJ: Wiley, 2014. ISBN 978-1118531648

- JavaScript Blog Mini Project
  - <http://lms.sabio.la/courses/prework-javascript/lectures/2264864>
- Bootstrap
  - <http://lms.sabio.la/courses/prework-javascript/lectures/2264921>
- JavaScript Book
  - <http://lms.sabio.la/courses/152326/lectures/2264880>
- jQuery
  - <http://lms.sabio.la/courses/152326/lectures/2264926>
  - <http://lms.sabio.la/courses/152326/lectures/2264933>
- JavaScript- Extra
  - <http://lms.sabio.la/courses/152326/lectures/2264862>
  - <http://lms.sabio.la/courses/152326/lectures/2264863>
  - <http://lms.sabio.la/courses/152326/lectures/2264938>

## **CLASS SCHEDULE**

This class schedule is for an introductory 12 week overview program in Full Stack Web Development. Classes will meet Tuesday, Thursday and Friday 1 p.m. to 5 p.m. (12 hours with instructor per week) with an additional 15 hours of student led time.

This is an interactive course where topics and lectures are accessed through the Sabio Wiki and google doc links that lead to more directly related material on the Web.

The following link leads to the material that is detailed below:

<http://lms.sabio.la/p/csc-1010/>

This section includes a general outline of each period.

Students will be given a logical order for the readings on certain varied topics covered in the Intro course by the instructor. Much of the Pre-work is meant to be done by the student on their own, with the help of instructors if necessary.

### **Phase 1 (WEEKS 1-2)**

During this period the instructor will teach students:

- Hardware and Software Requirements/Installation
- Installing Visual Studio.Net 2013 and Visual Studio 2013 Community Edition Update 4
- Basic programming in C# for Visual Studio.Net
- A practical introduction to HTML, CSS, JavaScript
- Understanding the need of code jQuery functions
- How to use Java Script and JQuery

- Handling events through jQuery

Assignments will reflect achievement of the topics described in this phase.

Reading Assignments:

As a first reading assignment students should follow these links to get acquainted with the material and where it is located. Students can gradually continue reading the support documents after reading the Initial Lectures:

- Reviewing [Hardware](#) and [Software](#) Requirements
- The Pre-Work Guide-Read Me
- An introduction to HTML, CSS, JavaScript
- Understand code jQuery functions
- Video Course: C# Fundamentals for Absolute Beginners: MS Virtual Academy:  
<https://www.microsoftvirtualacademy.com/en-US/training-courses/c-fundamentals-for-absolute-beginners-8295>  
[www.microsoftvirtualacademy.com](http://www.microsoftvirtualacademy.com)

Specific readings assigned by Instructor from the book:

**JavaScript and JQuery: Interactive Front-End Web Development** by Jon Duckett Link:

<http://amzn.com/1118531647>

**Pework 100**

On-line material in Codecademy on HTML/CSS/Javascript/JQuery:

<http://lms.sabio.la/courses/prework-full-stack-web-development-prereqs/lectures/2264688>

.Net/C#

<http://lms.sabio.la/courses/prework-c/lectures/2271863>

**Pework 200**

Read the JavaScript book discussed at the link below. Specific pages and chapters are called out:

<http://lms.sabio.la/courses/prework-javascript/lectures/2264880>

JQuery on CodeCademy:

<https://www.codecademy.com/learn/jquery>

jQuery Site

<http://lms.sabio.la/courses/prework-javascript/lectures/2264933>

**Pework 210**

On line material specified by instructor on basic HTML, CSS, and JavaScript in W3Schools Tutorials, references and examples

<http://www.w3schools.com/>

## Pework 300

JQuery Primary Functions

<http://lms.sabio.la/courses/prework-javascript/lectures/2264926>

Function vs Global Scope

<http://lms.sabio.la/courses/prework-javascript/lectures/2264936>

Bootstrap

Pework Extra: Only when you are done with the rest of the work.

<http://lms.sabio.la/courses/prework-javascript/lectures/2264921>

### Written/Project Assignments

Student has to take notes for each chapter of the JavaScript text and at the end of each chapter email them to instructor to evaluate student progress on each chapter.

Practice JQuery on CodeCademy :

<http://lms.sabio.la/courses/prework-javascript/lectures/2264933>

Pework Extra

Only when student is done with the rest of the work.

<http://lms.sabio.la/courses/prework-javascript/lectures/2264938>

Mini Blogging Pages

This project is encapsulated into 1 section within the given course. This is the first lecture in the section:

<http://lms.sabio.la/courses/prework-javascript/lectures/2264864>

## Phase 2 (WEEKS 3-6)

During this period the instructor will teach students:

- Visual Studio and basic elements of C# to write a Console application that accepts user input, writes output to the display, and performs basic conditional logic, looping and calculation.
- Static and Non-static classes and objects
- An introduction to ASP,Net MVC
- Solutions using ASP.Net Project: learning how to create the various action methods with MVC and views for editing the blog posts.
- To create controllers, action methods and views for editing data with MVC.
- To create a data access layer that communicates with the database students have been developing as part of the Database Pework.
- To Finish up the C# and Database prework
- To create a sample Blog application.

### Reading Assignments

The exercises for the following contents can be found here:

<http://lms.sabio.la/courses/prework-c/lectures/2264949>

- 210 C Sharp Basics
- 230 Static Classes
- 240 Non-Static Classes
- 310 MVC Intro
- 320 MVC Blog Part 1
- 330 MVC Blog Part 2
- 340 MVC Blog Part 3
- 350 - MVC Blog Part 4

### Written/Project Assignments

- Write a “Tip Calculator” console application:
- CSharpHomework01.docx
- Use the readings in MVC Blog Intro and MVC Blog Parts 1-4 to develop the Sample Blog.

### **Phase 3 (WEEK 7-9)**

During this period the instructor will teach students to:

- Install and use SQL Server Express 2014 and SQL Server Management Studio 2014
- Create stored procedures for Microsoft SQL Server databases
- Use functions that execute one or more SQL queries and procedural logic.
- Run scripts to create and populate a database using SSMS.
- Modify Data
- Store queries as stored procedures.

### Reading Assignments:

Pre-work-Database

SQL Server 2014 Express Installation Video: <http://lms.sabio.la/courses/lab-primary/lectures/2274366>

Pre-Work Guide for Database

120 Video Course “Database Fundamentals”: MS Virtual Academy

<https://www.microsoftvirtualacademy.com/en-US/training-courses/database-fundamentals-8243>

[www.microsoftvirtualacademy.com](http://www.microsoftvirtualacademy.com)

Prework 210- W3Schools Tutorials, references and examples on SQL and other relevant topics in the course.

<http://www.w3schools.com/>

<http://lms.sabio.la/courses/enrolled/153011>

- 310 Designing and Querying a Database
- 320 Data Modification:
- 330 Stored Procedures

- On-line Tutorial: Orientation on stored procedures.
  - <http://www.mssqltips.com/sqlservertutorial/160/sql-server-stored-procedure>
- 340 Stored Procedure Exercise
  - Get students started writing Stored Procedures that encapsulate the application's standard queries in the database as stored procedures.

### Written/Project Assignments

Java Script Project

### **Phase 4 (WEEKS 10-12)**

During this period the instructor will teach students to:

- Use JavaScript and JQuery for interactive front end development.
- Use JQuery Core.
- Apply JQuery functions such as Selectors, DOM Manipulation, Effects, and Events.
- Use plugins.
- Consider performance and code organization.
- Use JavaScript Hoisting.
- Finish up all practice exercises and projects with the help of the Instructor.

## ***Pre-Work Java-Script***

### Reading Assignments:

*JavaScript and JQuery: Interactive Front-End Web Development*, by Jon Duckett

- Pework – Bootstrap
- Pework – Codeacademy
- Pework – Javascript Book
- Pework 350 – jQuery
- Learn JQuery:
  - Follow materials here: <http://lms.sabio.la/courses/prework-javascript/lectures/2264933>
  - <http://learn.jquery.com/>
    - About jQuery
    - JavaScript 101
    - Using jQuery Core
    - Events
    - Effects
    - Ajax
    - Plugins
    - Performance
    - Code Organization
- jQuery API Documentation: <http://api.jquery.com/>

### *Written/Project Assignments*

#### JavaScript Blog Mini Project

- <http://lms.sabio.la/courses/prework-javascript/lectures/2264864> .

Students have to pass all the evaluations to continue on to the “Immersive Full Stack Web Development” course. Please see section “Evaluation Criteria”.

## OTHER COURSE AND UNIVERSITY POLICIES

### Attendance Policy

University policy states: “students are expected to attend all class sessions and, for online courses, participate in online discussions as required in the syllabus. If a student attends less than 80% of class sessions, the student will earn no credit for the course. If a student anticipates an absence for religious observance, work obligations, or any other reason, the student consults with the instructor before or during the first week of class to request an accommodation in the form of makeup assignments. In some cases, however, accommodation may not be possible if in the instructor’s judgment the absence would be disruptive to the learning process. In these cases the judgment of the instructor is final.” (See *AULA General Catalog*, <http://aulacatalog.antioch.edu/policiesregulationsandprocedures/academicpolicies>)

<http://aulacatalog.antioch.edu/policiesregulationsandprocedures/academicpolicies>

You are expected to attend all class sessions. If an absence is unavoidable, please contact me about a make-up assignment. This course program requires a highly intensive learning environment with accumulation of knowledge that depends on each previous assignment. Thus, the attendance policy reflects this need, and you will not receive credit if you miss more than 3 days or 9 hours of instructor-led class time.

### Letter grade equivalent policy

Letter grade equivalents will be given for this course. If you want a grade equivalent, please request it in writing by week two.

### Extra Credit Policy

Extra units are not allowed.

### Incomplete Policy

Incompletes are not available for this course. If you are not able to complete your assignments as required in the syllabus and as stipulated in the course meetings, you may not be able to receive credit for this course.

### Information Literacy and Research Requirements

All students are expected to develop an understanding of how to find and use resources appropriate for academic inquiry and scholarship. Although these are not skills that are required for the completion of this course, please be aware that students may meet with the AULA librarian should they need any help and support with research and for information literacy instruction.

### Student Conduct policy

Please provide expectations for student conduct in your class. EXAMPLE: “Respectful conduct is expected of students on the campus at all times, both inside and outside the classroom.” See *AULA General Catalog*, <http://aulacatalog.antioch.edu/policiesregulationsandprocedures/studentconduct/> for university policy.

## Plagiarism policy

University policy describes plagiarism as “the representation of someone else’s writing, graphics, research, or ideas as one’s own. Paraphrasing an author’s ideas or quoting even limited portions of the work of others without proper citation are also plagiarism, as is cutting and pasting materials from the Internet into one’s academic papers. Extreme forms of plagiarism include submitting a paper written by another person or purchased from a commercial source.” See *AULA General Catalog*, <http://aulacatalog.antioch.edu/policiesregulationsandprocedures/formsofprobationandtheirconsequences/>

### **\*Note about Avoiding Plagiarism in Computer Programs**

AULA does recognize that computer programming requires a specific interpretation of this policy to be consistent with accepted practices in the field. It is accepted that students will collaborate and share knowledge and code, including using code from other successful applications. However, it is expected that students make acknowledgements and, perhaps most importantly, be able to explain and defend the solutions created. If a student is unable to provide these types of learning evidence, this may be cause to suspect plagiarism and then the Antioch University policy regarding this may be followed. Additionally, we reference the “Guidance Notes on Plagiarism” from the Department of Computer Science at the University of Birmingham:

*Almost all computer programs contain many ideas borrowed from elsewhere. Many also contain short sections of actual code copied from elsewhere. For example, writing a section of program to create a new window on screen with a menu at the top of the window is often done by simply copying a few of lines of code from an example in a programming manual or textbook, either with or without a few minor changes. This is normally regarded as fair use and typically requires no acknowledgement.*

*Any more significant copying of code from elsewhere should be acknowledged, however. The acknowledgement can be put in comments within the program itself. Reference to the source of the original material should be made in the same way as in essays or other documents (except that it may not be possible to use italics or other font variations). Obviously, it is not possible to put sections of code in quotation marks to indicate that they have been taken directly from elsewhere. Instead, the comments should make it clear which sections of code have been copied from elsewhere. Equally, the comments should make it clear when the basic method has been copied from elsewhere, but changes made to the details.*

--<http://www.cs.bham.ac.uk/internal/studentinfo/plagiarism.htm>

## Reasonable Accommodation for Students with Disabilities

Antioch University is committed to providing reasonable accommodations to qualified students with disabilities in accordance with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 2008. Students who need to request disability accommodations should email or call Yaru Wang, disabled student services coordinator ([dss.aula@antioch.edu](mailto:dss.aula@antioch.edu) or 310-578-1080 x 209) at the outset of their enrollment, if possible, since reasonable accommodations are not retroactive.

## Sexual Harassment Policy

The Undergraduate Studies Division is firmly committed to each student's dignity and to eliminating all forms of sex discrimination and harassment of students. No student should have her or his learning experience at AULA contaminated by the experience of being treated as a sexual object by an instructor or any other employee. We strongly urge any student who believes that an Antioch employee has crossed the line to speak to your advisor, to the Undergraduate Studies Division leadership, the Provost, or the Director of Human Resources about your concerns.

Antioch University's policy "Title IX, Sex Discrimination, Sexual Harassment, and Sexual Violence" provides definitions of prohibited and inappropriate behaviors, the process for reporting and investigating complaints, and the sanctions levied against those employees or students found to be in violation of these policies. This policy can be found in the Antioch University Resource Archive at [http://aura.antioch.edu/policies\\_400\\_6x/12/](http://aura.antioch.edu/policies_400_6x/12/).

Additionally, please visit the link below for Antioch University's policy on dual relationships:

[http://aura.antioch.edu/policies\\_400\\_6x/11/](http://aura.antioch.edu/policies_400_6x/11/)

### **Antioch University Policies:**

Antioch University is committed to building a vibrant and inclusive educational environment that promotes learning and the free exchange of ideas. Our academic and learning communities are based upon the expectation that their members uphold the shared goal of academic excellence through honesty, integrity, and pride in one's own academic efforts and respectful treatment of the academic efforts of others.

All students are expected to comply with Antioch University policies, including the Title IX Sexual Harassment and Sexual Violence Policy and the Student Conduct Policy.

To access academic, student, and other university policies are available online:

[http://aura.antioch.edu/au\\_policies/](http://aura.antioch.edu/au_policies/)