AP Computer Science Principles
Course Syllabus

"Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination."
Albert Einstein

"The spread of computers and the Internet will put jobs in two categories. People who tell computers what to do, and people who are told by computers what to do."
Marc Andreessen

"Everybody should learn to program a computer, because it teaches you how to think."
Steve Jobs

Course Information
Overview
AP computer science principals is a college level computer science course, where students have the opportunity to learn advanced computer science concepts and earn college credit while still in high school. The course focuses on blending hands-on coding practice with other elements of computing knowledge, including digital ethics and computing security.

Benefits of studying computer science
In the 21st century, much of the modern world is increasingly controlled by computers. Regardless of your chosen career path, your job will most likely involve interacting with a computer. If your prospective career involves any form of research or engineering, you will most likely also have to program a computer. Learning these skills in high school allows you to get a heard start on other college students, and save time and money when selecting college courses. Knowing these skills will also give you creative problem solving skills that will be helpful regardless of your future field of study.

Grading policy
Assignments given will be in the following categories and will be have the following weights:
- Unit tests - 37.5%
- Homework projects - 37.5%
- Multiple choice quizzes - 25%

Late work will be loose 0.023% per minute late. This means that if you submit an assignment 1 day late, you will lose 33% of its grade, and if you submit an assignment 3 full days late, you will receive no credit. If adjustments or extensions are needed, they will be evaluated on a case-by-case basis with a focus on maximizing student success and continued learning.

Programming projects will be graded as follows:
- 40% of credit will be awarded for having a program that compiles successfully
- 30% of credit will be awarded if your program produces the specified output
- 15% of credit will be awarded if your program complies with the requirements in the style guide
- 15% of credit will be awarded if your program demonstrates creative and innovative development strategies and methodologies

Homework
The primary homework assignments in this class will be in the form of week long programming assignments. These assignments will be designed to allow you to creatively use the programming skills that you have built throughout the course, in a cumulative manner. Each assignment will ask you to use your coding skills to build a program that accomplishes some real world goal. When this is not applicable, as in the case of conceptual units, you will be assigned written assignments, where you will be asked to use the skills you have learned to describe how you would solve a real world problem, or implement a technology.

Tests and Quizzes
The primary means of assessment in this class will be AP style multiple choice tests and quizzes. You will receive a quiz at the end of each section, and you will receive a test at the end of each unit. The short quizzes are designed to not require anything more than a cursory read though of your notes. If you find yourself having to cram for these quizzes, you may be having trouble with the section's material. The end-of-unit tests will be more extensive, and will require review.

Absentee policy
All homework and textbook work will be posted on the class server, so that absent students can complete their projects and homeworks. Absent students will be provided with extensions on work if they are absent, but they will still be expected to complete them. I will make every effort to provide resources to absent students, but students are strongly encouraged to come to class for tests and quizzes, and to receive lessons. If students are late or absent too much, they run the risk of falling behind on essential content that they are going to need for future lessons.

Textbook information
Students in this class will be using the textbook *Computer Science Illuminated, 7th edition* by Nell Dale and John Lewis for the majority of the course. In the units analyzing the effects of computing on society, some sections of *Blown To Bits: Your Life, Liberty, and Happiness After the Digital Explosion* by Harry Lewis, Ken Ledeen, Hal Abelson, and Wendy Seltzer will be assigned. Textbook copies will be provided for all students. Students will be expected to read the textbook sections relevant to current classwork and to refer back to past chapters when needed. Students will also be expected to use practice questions and other textbook resources to practice difficult material.

AP Exam Information
Exam content
The AP computer science principals exam is composed of two parts. The first part consists of a 70 question multiple choice exam, which students will have 120 minuets to complete. This exam will include 8 questions where students are required to select multiple answer choices to receive full credit. The second part of the exam is a long-term performance task that must be completed over 12 hours of class time. For this part of the exam, students will be expected to include many elements of programming that have been covered over the course of the class to create a program that preforms some form of real-world function. Part one of the exam will be worth 70% of the exam credit, and part two will be worth 30%.

AP Big Ideas
AP has defined several key ideas for this course, which will shape the direction and flow of the curriculum for the entire year, and form the backbone of the AP exam content. They are listed below, along with their percentage on the AP examination. For more information on each of the following,

refer to the AP course and exam description document pertaining to this course.

- Creative development: 10-13% of the exam
- Data: 17-22% of the exam
- Algorithms and programming: 30-35% of the exam
- Computer systems and networks: 11-15% of the exam
- Impact of computing: 21-26% of the exam

AP computational thinking practices

The following are the AP computational thinking principals, which have been woven into every aspect of the course's curriculum. You should strive to consider how these practices impact every program that you write throughout the course of this class.

- Computational solution design
- Algorithm and program development
- Abstraction in program development
- Code analysis
- Computing innovations
- Responsible computing

Course Content

Unit 0 - The basics of computer programming

This unit will focus on building basic computer skills, including how to access and use the course's programming system, the parts of a computer, and some basic computer history. This unit will conclude with and introduction to the basics of writing and executing python code.

Section 0 - Introduction and getting set up

This section covers the basics of the Linux programming environment, and an introduction to why we study computer science.

Textbook: Computer Science Illuminated chapters 1 and 11

Unit project: Linux scavenger hunt

> In this project, students will be asked to navigate the linux directory system to find certain files, manipulate these files using simple linux tools, and then submit the new file using the course ware. [P1]

Lessons:

- Why study computer science?
- Introduction to the Linux computer environment
- Practical Linux 1
- Practical Linux 2
- Practical Linux 3

Section 1 - History of computing (IOC) [P5]

This section will focus on reviewing the history of computing from the early 1950s to the current day. This unit will use the overview of computing history and technology to analyze and draw conclusions about the current state of computing technology and ethics.

Textbook: Chapter 1 and 10 in Computer Science Illuminated

Section project: Analyzing a computing advance paper (IOC) [P5]

> Students will choose one of the 3 major historical computer advances, and will explain how this computer innovation impacted future computing innovations. Students will also be asked to analyze the impacts that the advance had on privacy and society as a whole.

Lessons:

- Early computing and batch processing
- The IBM system 360 and time sharing
- Bell labs, UNIX, and the free software movement
- The microcomputer revolution and the IBM PC
- The Internet age

Section 2 - How does a computer work? (CSN) (AAP) (DAT) [P3] [P5]

This section will analyze the inner workings of a computer at the hardware level. Students will learn how CPUs process instructions, manage memory, communicate with peripheral devices, and encode and decode binary data. Students will also learn basic information about binary and hexadecimal number systems.

Textbook: Computer Science Illuminated chapters 2, 4, 5, 10

Section project: Building a mock CPU (DAT) (CRD) [P1] [P3]

Students will use a library of components to describe a computer system from scratch. They will have to describe a CPU instruction set, memory system, IO system, and user interface. Students will have to describe the benefits and drawbacks of the computer system that they designed, and what its commercial applications may be.

Lessons:

- The hardware layer
- I/O, memory and binary code
- CPU instructions and low level programming
- The operating system layer
- Introduction to abstraction

Section 3 - Introduction to programming (AAP) [P1] [P2]

This section will introduce students to the basics of programming that they will need for future units. Students will learn the foundations of python code including how to print text, assign integers, and get input from the user.

Textbook: Computer Science Illuminated chapter 7

Section projects: Hello, World (AAP) [P2]

For this project, students will be expected to write a simple program that retrieves the name of a student, and then says hello to the student. Student's programs must compile, and be submitted through the course's programming environment.

Lessons:

- What is python? How do I write it?
- Introduction to variables and data types
- Getting user input
- Introduction to conditional logic
- Introduction to loops

Unit 1 - Basics of algorithms and programming

This unit will introduce students to the basics of programming algorithms and computer programming. In this unit, students will learn the basics of programming discreet algorithms, which is the foundation of all future computer programming skills. By the end of this unit, students will be able to write programs that use conditional logic, repetition, and then debug these programs for speed and efficiency.

Section 0 - Algorithms and computing (AAP) [P1] [P2]

This section will introduce students to functions and conditional statements. By the end of this unit, students will be able to write programs that incorporate loops and if statements.

Textbook: Computer Science Illuminated chapters 6 and 7

Unit project: Prime finder (AAP) [P1] [P2]

> Students will be assigned to build a simple program that finds an arbitrary number of prime numbers. The program must get user input about how many primes to find, and output an array containing that number of prime numbers. Programs will be required to use an array, a loop, and an if statement

Lessons:

- Arrays 1
- Loops 2
- Simple math functions
- Advanced decision making
- Functions

Section 1 - Data Structures (DAT) [P1] [P3]

This unit will cover arrays, as well as going into more depth on standard primitive data types. Students will also receive a brief introduction to strings and other more complex data types.

Textbook: Computer Science Illuminated chapters 2 and 3

Section project: String tool (DAT) (AAP) [P1] [P2] [P3]

> Students will be required to write a program that gives users multiple options about how to manipulate a target string. Student's programs must be able to invert, split, join, and compare strings. Student's programs must also be able to count the number of Z characters in the string. All operations should be implemented as separate functions.

Lessons:

- Types of primitives in computers
- Strings
- Arrays 2
- Arrays 3
- Dictionaries and ordered data

Section 2 - Algorithm design, time complexity, and efficiency (CRD) (AAP) [P1] [P2] [P3]

This section will introduce students to the basics of designing algorithms instead of just programming them. Students will learn about how to design efficient algorithms, and what the limits of computing are. This unit will also cover the basics of debugging programs for logical errors.

Textbook: Computer Science Illuminated Chapter 7
Section project: Combined project (sections 2 and 3) - Building a custom sorting algorithm [P1] [P2] [P3] (CRD) (AAP) (DAT)

> This will be a combined project across sections 2 and 3. In this section, students will pre-plan the algorithm that they plan to build in chapter 3. They will be required to draft a flowchart depicting the algorithm that they will implement to sort the list of numbers. The flowchart must include all conditional statements, loops, calls to storage, and I/O operations. Students will then have to analyze the program that they wrote. They will have to note the program's time complexity, and note the practical and theoretical limits of their algorithms.

Lessons:

- State 1
- Limits of computing
- Time complexity
- Algorithmic efficiency
- Debugging for logic errors

Section 3 - Deep dive into sorting and searching algorithms (CRD) (AAP) [P1] [P2] [P3]

This section will continue on the themes established in the last section by examining the theoretical principals of algorithm design in the practical context of the common searching and sorting algorithms that most programs implement. This unit will focus on how to code the design patters discussed in the last section. Students will also learn how to consider how program efficiency effects real-world tasks, and will learn how to optimize their programs to run faster.

Textbook: Computer Science Illuminated Chapter 7
Section project: Combined project (sections 2 and 3) - Building a custom sorting algorithm [P1] [P2] [P3] (CRD) (AAP) (DAT)

> This is a continuation of the project started in section 2. In this project, students will implement the flow charted programs that they designed in section 2's project. Students will then work on optimizing the programs that they wrote using the techniques that they learned in this section. Finally, students will compare the sorting algorithm they built to one that was explored in depth in this section.

Lessons:

- Sorting
- Searching
- Comparing different algorithms
- Real world efficiency concerns
- Debugging for speed and efficiency

Unit 2 - Data

Section 0 - Data abstraction (DAT) (AAP) [P2] [P3]

This section will go into more detail about how data is abstracted within a computer system. Topics introduced earlier such as hexadecimal and binary number systems will be expanded upon, and students will learn how these concepts relate to the compression of data. This chapter will also introduce students to data abstraction at higher levels, such as objects and user-defined data structures.

Textbook: Computer Science Illuminated Chapters 2, 3, and 8
Section project: Binary-Hexadecimal-Text Converter (CRD) (AAP) (DAT) [P1] [P2]

> In this project, students will build a program that converts hexadecimal digits into ascii characters. Students will be required to implement a subset of the ascii character table, and programs must be able to parse stored and real-time input.

Lessons:

- Binary data storage
- Hexadecimal abstraction of binary
- Data compression overview
- Compression algorithms
- General abstraction

Section 1 - Database theory (DAT) (CRD) [P1] [P3] [P5]

In this unit, students will be introduced to the theoretical concepts behind relational databases. This unit will cover the benefits of using a database, how they can be used to extract meaning from data, and how they can be used to manipulate data.

Textbook: -
Section Project: Database selection

> In this project, students will be presented with a variety of hypothetical applications, and students will be required to select which kind of database would be the best fit for that particular application. Students will then be required to describe how that database could be implemented in that application, and the benefits of doing so.

Lessons:

- What is a database?
- Types of databases
- Relational databases
- Keys
- Normal form

Section 2 - SQL and database programming (DAT) (CRD) (AAP) [P1] [P3] [P5]

This unit will provide students an introduction to the basics of database programming. Students will learn how to make SQL queries to select, add, update and delete data. Students will also learn about joins, tables, and keys.

Textbook: -
Section Project: (Combined Project sections 1 -2) Concert data analysis (DAT) (CRD) (AAP) [P1] [P2] [P5]

> In this project, students will be given information representing various statistics surrounding attendance at a music concert by a particular artist. Students will be assigned to construct a

database program that can create, update, and remove data from this database. Students will also have to construct a flexible selection system that will allow them to program in selection queries based on other variables or events in the program.

Lessons:
- Selection queries
- Creating and updating records
- Manipulating tables
- Keys
- Joins

Section 3 - Data and its interpretation (DAT) [P1] [P2] [P3] [P5]

This section will show students the real utility of programs as more than a method of crunching numbers. Students will learn how enterprises and scientists use computers to extract meaning from data, and the technical and architectural challenges involved in doing so. Students will practice planning how to build programs that can not only process data and extract meaning from it, but also do so at scale.

Textbook: Computer Science Illuminated Chapter 12
Section Project: (Combined Project sections 1 -2) Concert data analysis (DAT) (CRD) (AAP) [P1] [P2] [P5]

In the second part of this project, students will use the programs that they built in the first part of the assignment to generate a number of actionable reports about the concert data. Student's analysis programs must use at least one join function, one aggregation function, and one key operation.

Lessons:
- What is data?
- Information management systems
- Extracting information from data - 1
- Extracting information from data - 2
- Generating actionable reports

Unit 3 - Software development and object orientation

Section 0 - Software development practices (CRD) [P1] [P4]

This section will focus on the knowledge students will need to know to manage large software projects effectively. This unit will cover collaborative development, documentation, and more information about how to debug programs that have multiple functional algorithms. Students will also learn about development methodologies including the iterative and waterfall design processes. Lastly, this unit will provide a brief introduction to object orientation, functional programming, and other aspects of state management.

Textbook: Computer Science Illuminated Chapters 8 and 9
Section project: Writing a software specification (CRD) (IOC)[P1] [P2]

> For this project, students will be assigned to interview a "client" and use the information gained from this interview to draft a software specification that could be used for further development later on. Student's specifications must contain all the functionality requested by the "client", while presenting enough information for the "developers" to build a full program out of it.

Lessons:

- Collaborative development
- Iterative development to a user specification
- Debugging for user specifications
- Managing large projects – State 2
- Managing large projects – Programming Paradigms

Section 1 - Enterprise software (DAT) (IOC) (CSN) [P1] [P2] [P5]

This unit will focus on how software development practices change with the scale of the software being developed. Students will learn about how software can be engineered for fault tolerance, large loads, and other complications that can arise when software is being used by thousands or even millions of people. This unit will also cover how software is used by these large institutions to process data.

Textbook: Computer Science Illuminated chapter 12
Section Project: Combined Project (Sections 1-3) Banking Application (DAT) (CRD) (AAP) [P1] [P2] [P3]

> In the first part of this project, students will plan out how they will create a digital banking program using an object oriented design model. Students will create an object map that shows the relationships between the parts of the program that they are going to construct, and a flowchart that depicts the logical flow of the program. Students plans must be for a program that can manage deposits, withdrawals and transfers between accounts.

Lessons:

- What is software actually used for?
- Information management
- Running at scale
- Fault tolerance
- Documentation and practice

Section 2 - Principals of object oriented design (CRD) [P1] [P3]

This unit will introduce students to the basics of designing object oriented programs. Students will learn about the fundamental principals of object oriented design, and concepts like inheritance and

polymorphism.

Textbook: Computer Science Illuminated Chapter 9
Section project: Combined Project (Sections 1-3) Banking Application (CRD) (DAT) [P1] [P2] [P3]

> In part two of the project, students will implement the plans that they made in part one of the project. Students will develop and debug their programs to produce a fully functional banking application that meets the requirements of the project and conforms to the plan that they developed from part one.

Lessons:

- What is an object
- Inheritance
- Polymorphism
- Object oriented design
- Object orientation drawbacks

Section 3 - Object oriented programming (CRD) (AAP) [P1] [P2] [P3]

In this section, students will build on the knowledge they learned in the last section, and begin implementing object oriented design principals. They will learn how to add constructors, methods, and variables to their objects. This unit will also introduce a deep dive into the string object, to show what object oriented design is capable of.

Textbook: Computer Science Illuminated Chapter 9
Section project: Combined Project (Sections 1-3) Banking Application (CRD) (AAP) [P2] [P3] [P4]

> In the final part of the project, students will implement a reporting feature that can derive business analytics from the program in question. Students will be required to develop and integrate reports that they think would be useful to a business using the software in question. Reports must be printed to a text file with appropriate formatting.

Lessons:

- Constructors
- Accessing and private variables
- Methods
- The string class 1
- The string class 2

Unit 4 - Simulation, supercomputing, and networks

Section 0 - Simulations and modeling (CRD) (AAP) (CSN) (IOC) [P1] [P3] [P5]

This unit will apply the concepts covered in the last section to teach students how to use supercomputing systems to build digital models of real-world phenomenon to save time and produce more accurate results. Students will learn which types of applications are good candidates for modeling, the benefits and drawbacks of using simulations, and the factors that must be considered when constructing models.

Textbook: Computer Science Illuminated Chapter 14
Section project: Combined project (Sections 1 and 2) - Checkout line simulation (CRD) (AAP) [P1] [P2]

> Students will be tasked with building a simulation of a store checkout line. Student's simulators must be able to find the average time that a customer spends waiting, based on the time an employee takes to handle a transaction, and the number of customers in the line. Students will be required to determine the best configuration for 3 grocery stores based on customer count, budget, and allowed waiting time.

Lessons:

- What is a simulation?
- How can we simulate the real world?
- Pros and cons of simulations
- Implementing simulations in computers
- Computing requirements of simulations

Section 1 - Supercomputing (CRD) (AAP) (CSN) (IOC) [P1] [P5]

This unit will cover some of the alternative computing approaches that are needed when the amount of data that needs to be processed is truly vast. Students will learn what makes a computer go fast, the kinds of applications that need such performance, and the benefits and drawbacks of parallel computing systems.

Textbook: Computer Science Illuminated Chapter 14
Section project: Combined project (Sections 1 and 2) - Checkout line simulation (CSN) [P1] [P4] [P5]

> In this part of the project, students will be required to analyze how their projects would run on various types of supercomputers. Students will describe what changes might be necessary to the program to maximize its performance on said supercomputing models. Students will also be tasked with designing an ideal supercomputing model for the problem being analyzed.

Lessons:

- What makes a computer fast?
- History of supercomputing
- Parallelism
- Grid Computing
- Applications of supercomputing

Section 2 - Networks (IOC) (CSN) [P1] [P3]

This section will introduce students to the technical details of networking. They will lean how networks are built, the different types of networks, and how data is routed and switched along the different types of network. Students will also learn the advantages of connecting their devices to a

network, and programming for networked environments.

Textbook: Computer Science Illuminated Chapter 15
Section Project: Network design (CSN) [P1] [P5]

In this project, students will be required to build a network map that describes an enterprise network which meets the specifications given. Networks must include at least 3 sites, a server room, and a wireless LAN. Students will be responsible for assigning networking devices to locations within the network, assigning subnets and determining the best link speed for each application.

Lessons:

- Why network? & The history of networking
- TCP/IP
- Routing
- Network architecture and devices
- Networked applications

Section 3 - The Internet and networked applications (CSN) (IOC) [P1] [P3] [P5]

This section will focus on how networking impacts applications and application development. Students will learn about how services like E-mail and world wide web work, and how they can program their own applications to interface with this worldwide network of devices. Lastly students will learn about the architecture of the Internet as a whole, including the basics of services like DNS and BGP.

Textbook: Computer Science Illuminated chapter 16
Section Project: Internet chat program (CRD) (AAP) (CSN) [P1] [P2] [P3]

In this project, students will be assigned to create a simple chat protocol that they will then implement in a simple program using TCP sockets. Students will have to design a protocol that allows for reliable transmission of a text message, information about the origin and receiver, and session control information. Students will then have to implement their protocol in a program using TCP sockets.

Lessons:

- Architecture of the Internet
- DNS
- Internet applications - The world wide web and Email
- Internet applications - Programming your own
- The Internet of things

Unit 5 - Security

Section 0 - Encryption (AAP) (CRD) (DAT) (IOC) [P1] [P2] [P5] [P6]

In this unit, students will learn the basics of encryption algorithms. Students will learn why encryption is needed, and where it can be found. Students will also be taught what encryption algorithms are available, and receive a basic overview of how they work.

Textbook: Computer Science Illuminated chapter 17, Blown to Bits chapter 5
Section Project: (Combined project: Sections 1 - 2) Custom Encryption System (CRD) (AAP) (DAT) [P1] [P5] [P6]

> In this project, students will be tasked with building a simple encryption algorithm. Students will be assigned an application, and will be required to research and implement a cipher of their choosing. Student's programs must be able to meaningfully encrypt their assigned data in a reversible manner.

Lessons:

- What is encryption and why is it necessary?
- An overview of some simple encryption algorithms
- Private key encryption
- Public key encryption
- Code-breaking basics

Section 1 - Program security [P1] [P2] [P6]

This unit will introduce students to common vulnerabilities found in program code, and the effects they can have on applications and programs in the real world. Students will also learn how these security holes can be repaired, and prevented in the first place.

Textbook: Computer Science Illuminated chapter 17
Section project: (Combined project - Sections 1 - 2) Custom Encryption system (CRD) (AAP) (DAT) [P1] [P2] [P4] [P6]

> In this portion of the project, students will have additional time to finish coding their encryption systems, and will also be required to debug their programs for potential security vulnerabilities.

Lessons:

- Buffer overflows
- Injection attacks
- Authentication systems
- Unsecured user data
- Debugging for security

Section 2 - Internet Security [P1] [P2] [P6]

This unit will focus on common Internet security threats, including phishing, ransom-ware, spam email, and data breaches. Students will learn about the origins of these common threats, the methods by which they operate, and how to protect themselves against them.

Textbook: Computer Science Illuminated chapter 17
Section Project: Malware Removal (CRD) (AAP) (IOC) (CSN) [P1] [P6]

> For this project, students will be required to remove malware from a windows virtual machine that has been prepared for this purpose. Students will be required to remove all malware from

the target system using publicly available tools. Students will then be required to write a report describing the types of malware that were installed on the system, and the potential damage that could have been caused if the malware had continued to remain on the system.

Lessons:
- History of computer viruses
- How computer viruses actually work
- Botnets
- Email spam and phishing
- Wireless security

Section 3 - Hacking and data breaches (CSN) (IOC) [P5] [P6]

This section will focus on the methods that hackers use to gain access to systems, and the consequences of a data breach. Students will analyze data breaches to determine how hackers accessed the systems, and what information they stole. Students will learn about the devastating consequences of a breach, and how to prevent one in the future.

Textbook: Blown to Bits chapters 1-2
Section Project: Analyzing breach report (CSN) (IOC) [P5] [P6]

In this project, students will be assigned a breach report of a fictional company, describing a data breach. Students will be assigned to figure out what data was stolen, the effects that this might have on the company, and what should be done about it. Students will also be required to determine how the hackers broke into the company and data in question.

Lessons:
- Social Engineering
- Ransom-ware attacks
- Data breaches
- Analyzing some major hacks - part 1
- Analyzing some major hacks - part 2

Unit 6 - Computing innovations

Section 0 - Effect of computing (IOC) [P5]

This section will focus on the impacts of computing innovations on society. This unit will show students the impacts of the products they produce have on the world around them. Students will also explore the responsibility that they have for the computing innovations and devices that they create. Lastly, this unit will explore how the purpose and use of a computing innovation can shift over time from what it was originally meant to do.

Textbook: Blown to Bits chapters 3-4
Section Project: Analyzing purpose drift of a computing innovation (IOC) [P5]

> In this project, students will analyze how the use of a computing innovation could change over time. Students will describe how the use of the computing innovation could change over time, and the positive and negative effects that this may have on its users, and society as a whole.

Lessons:

- How does innovation impact the world?
- Positive effects
- Negative effects
- Purpose drift – How do you plan for how users will use your application in the future?
- How are you responsible for your software?

Section 1 - Studying some cutting edge computing innovations (IOC) [P5]

This unit will focus on studying the effects and impacts of several recent high tech innovations. Students will analyze the effects that these innovations have on society, and how the negative effects may be remediated. Lastly, students will also analyzes the data and information that the program uses, and how this may pose a security risk.

Textbook: Blown to Bits chapters 4-5
Section Project: Analyzing the data used by a computing innovation (IOC) [P5] [P4] {CI 1, B}

> In this assignment, students will be required to analyze how a computing innovation processes data to produce a useful result. Students must describe how the program acquires data, how it stores it, how it processes it, and then how it outputs the data. Then students will describe how the program uses this data to produce a useful result or output.

Lessons:

- AI
- Quantum computing
- Automation
- Deep dive – crowd-sourcing 1
- Deep dive – crowd-sourcing 2

Section 2 - Privacy and data collection (IOC) [P5] [P6]

In this section, students will learn about the effects that data collection has on the privacy and security of a program and its users. Students will explore the pros and cons of ‘big data’ and widespread data collection in a program. They will learn about how this data can be used and misused by a variety of different people who may have interest in it. Additionally, students will also learn about copyright and fair use when using digital assets and media. Lastly, students will learn about open source software, and code sharing.

Textbook: Blown to Bits chapters 6-7
Section Project: Determining a privacy policy (IOC) [P1] [P5] [P6] {CI 2, C}

In this project, students will be assigned to investigate a computing innovation and determine what data the company should keep, and how it should be stored. Students will have to balance multiple factors in their assessment, including legal risks, ethical issues, financial limits, and technological constraints. When finished developing their plan, students will swap plans with a partner, who will identify data privacy and security issues in the plan. In this phase of the project, students will determine what might happen in the case of multiple types of cyber-attacks, and will then evaluate the plan using a provided rubric.

Lessons:

- Privacy in the Internet age
- What data do you generate and what data is collected?
- Data Breaches and leaks
- Copyright
- Open source and fair use

Section 3 - Digital divide and computing bias (IOC) [P5]

In this section, students will explore the digital divide, and how bias impacts algorithms and computers. Students will explore the differences in computing resources available by geography, socioeconomic status, and many other factors. They will then use this information to analyze how computing innovations and legal polices may have different effects depending on where they are implemented. Students will also explore how algorithms can become biased, the effects that this may have, and how it can be resolved.

Textbook: Blown to Bits chapters 8-9
Section Project: Analyzing the impact of a computing innovation (IOC) [P5] {CI 3, A}

In this section, students will be assigned a computing innovation, and they well be required analyze how it will impact 5 geographically and socioeconomically distinct groups. Students will be required to determine how the innovation would impact each of these groups, for better or worse. Students will then be required to redo this analysis if it was later determined that the algorithm implementing the innovation was biased in a specified manner.

Lessons:

- The digital divide - Socioeconomics
- The digital divide - Other factors
- What is bias and how does it enter algorithms?
- No win situations: Accuracy vs bias
- How do you prevent bias in computing?

Unit 7 - Create Performance Task

After completing unit 5, students will be given twenty, 45 minuet class periods to complete their performance task. Before the allotted time commenced, students will be given a detailed description of the task's requirements, and grading rubric. Students will also be shown several completed examples of performance tasks, and their topics will be reviewed. Students will then be allowed to work on their assignments, before they are guided through the submission process on AP digital portfolio.

Unit 9 - Web Applications

Section 0 - Basics of Web Applications (CRD) (AAP) (CSN) [P1] [P2] [P5]

In this unit, students will learn in-depth about the HTTP protocol and the functioning of a web server. Students will learn how data is transmitted and displayed to the client, and how web browsers enable advanced functions like reactivity and interactivity. Lastly, students will learn about the model view controller paradigm.

Textbook: -

Section Project: (Combined Project: Sections 0-1) Hello Website (CRD) (AAP) [P1] [P2] [P5]

> In this project, students will copy the basic hello website into their web server folders, and then they will start up their web servers and navigate to their page in a web browser. Students will then use their browser's inspect element feature to determine how the site's elements were transmitted to the client, and how they are displayed.

Lessons:

- The HTTP server
- The web browser
- Server and client side website programming
- Website database connection
- The model view controller paradigm

Section 1 - HTML and CSS (AAP) (CRD) [P1] [P2] [P5]

This section will cover the basics of building static websites in HTML and then styling them in CSS. Students will learn how to lay out and style basic text, link and image elements. They will also receive a brief overview of interactivity using CSS selectors and in-line JavaScript.

Textbook: -

Section Project: (Combined Project: Sections 0-1) Hello Website (CRD) (AAP) [P1] [P2] [P5]

> In the second half of this project, students will extend the Hello website from part one to include multiple pages, a menu, some images, and styled elements using CSS. Student's websites must include a centered flex-box, imported style-sheet, and image.

Lessons:

- HTML basics
- The DIV and Flex-box
- CSS overview
- Importing content and using simple frameworks
- A brief overview of JavaScript

Section 2 - Back-end Applications (CRD) (DAT) (AAP) (CSN) [P1] [P2] [P5]

In this unit, students will learn the basics of connecting websites to information systems. Students will learn how to connect their websites to databases, how to retrieve information from web requests, and how to transmit this information to the display layer of a website.

Textbook: -

Section Project: (Combined Project: Sections 2-3) Concert booking web application (CRD) (DAT) (AAP) (CSN) [P1] [P2] [P5]

> In this project, students will connect a web application to the concert information database program from unit 2. Students will construct a back-end HTTP API that can be queried to re-

trieve or update booking information from this database. Student's programs must securely pass queries though to the database to allow for the addition of new bookings, the retrieval of existing bookings, and the removal of unwanted bookings. Student's programs must use both GET and POST requests.

Lessons:
- What is an API
- Getting data
- Returning data
- Connecting a database
- Routing requests

Section 4 - Front-end Applications (CRD) (AAP) (CSN) (IOC) [P1] [P2] [P5] [P6]

In this unit, students will learn how to use a JavaScript framework to construct a front-end for a HTTP api. Students will learn how to use JavaScript to dynamically retrieve and display data, and allow users to interact with the website. Students will also learn how to make their website reactive, and allow it to function on both mobile and desktop platforms. Lastly, students will learn how to secure their web applications against intrusion.

Textbook: -
Section Project: (Combined Project: Sections 2-3) Concert booking web application (CRD) (AAP) (CSN) (IOC) [P1] [P2] [P5] [P6]

In this project, students will construct a front-end for the back-end API that they built in the last section. Students front-end's must be responsive, working on both mobile and desktop platforms, and allow students to use all aspects of the API that they built in the last section. Student's applications must also be secure, and free from simple website vulnerabilities.

Lessons:
- JavaScript frameworks
- Reactivity
- Sending requests
- Processing data
- Web application security