Summary Course Description

Computer Science Principles (CSP) is a PLTW course that implements the College Board's AP CS Principles framework. The course does not aim to teach mastery of a single programming language but aims instead to develop computational thinking, to generate excitement about the field of computing, and to introduce computational tools that foster creativity. The course also aims to build students' awareness of the tremendous demand for computer specialists and for professionals in all fields who have computational skills. The course aims to engage students to consider issues raised by the present and future societal impact of computing. Students practice problem solving with structured activities and progress to open-ended projects and problems that require them to develop planning, documentation, communication, and other professional skills. They work in teams to combine team members' perspectives, skills, and knowledge to address a complex problem. This team approach gives students opportunities to describe, explain, and justify the design and appropriateness of their computational choices, to develop, analyze and describe computational artifacts and the behavior of programs, to communicate the meaning of knowledge discovered in data, and to use simulations to understand, predict and communicate about natural phenomena. Problems aim for ground-level entry with no ceiling so that all students can successfully engage the problems. Students with greater motivation, ability, or background knowledge will be challenged to work further.

CS Principles: Computational Thinking Practices and Big Ideas

Students develop the six Computational Thinking Practices and achieve enduring understandings of the seven Big Ideas by collaborating to create solutions to problems. They evaluate and communicate solutions and connect computing to all areas of work and society. In considering any problem, students build the habit of considering five ways in which computing can be applied to a problem.

Computational Thinking Practices:

- Connecting Computing
- Creating Computational Artifacts
- Abstracting
- Analyzing Problems and Artifacts
- Communicating
- Collaborating

Computer Science Big Ideas:

- Creativity
- Abstraction
- Data and Information
- Algorithms
- Programming
- Internet
- Global Impact

Applying Computing to a Problem

- Express ideas with creativity
- Safely and effectively use the Internet
- Collect, visualize, analyze, and communicate data
- Model and simulate
- Create and improve algorithms and automate

Computer Languages Used

During the unit on algorithms, students are introduced to programming with *Python*®, which is the primary language used in the course. During the unit on the Internet, students also work with HTML, CSS, JavaScript, PHP, and SQL.

Learning Objectives

The course follows the Understanding by Design model in which each lesson is designed to produce evidence that students have achieved specific course objectives. Course materials are explicitly aligned to objectives at the lesson level and at the activity level, including PLTW skills and knowledge objectives; CS Principles learning objectives and essential knowledge; and objectives specified in NGSS, CCSS, CSTA, and other standards.

College Credit Options

- All students **must** take the course end of year exam given by PLTW, regardless of whether or not they choose this option for college credit. Any student who earns a score of 6 or higher (out of 9) on this exam is eligible for 3 college credits through the North Dakota State University. These credits may be transferred to other colleges and universities.

- Students have the option to pursue earning college credit through the College Board by taking the Advanced Placement (AP) Examination. In that case, in addition to the course end of year exam, this option requires the completion of three College Board assessment components. There is a Performance Task that must be completed in a proctored environment without assistance or input from a teacher. Students must also sit for the multiple choice AP exam in May. Please note that the results of these assessments will not be factored into the student's course grade.

Computer Science Principles Curriculum

Unit 1: Algorithms, Graphics, and Graphical User Interfaces

The goal of Unit 1 is to excite students about programming and to build their algorithmic thinking and ability to use abstraction. Student creativity is emphasized as they work with the Python® programming language to tell graphical stories and explore various development environments and programming techniques. Students create original code and read and modify code provided from other sources. An Agile software development process is emphasized and personal, professional, and collaborative skills take center stage. Students debate policy questions about the ownership and control of digital data and examine the implications for creative industries and consumers. In this unit students begin their exploration of career paths tied to computing. Unit 1

1.1 Algorithms and Agile Development
1.2 Mobile App Design
1.3 Algorithms in Python
1.4 Images and Object-Oriented Libraries
1.5 GUIs in Python (optional)

Unit 2: The Internet

The goal of Unit 2 is for students to have a more concrete understanding of the Internet as a set of computers exchanging bits and the implications of these exchanges. Students use PHP and SQL to structure and access a database hosted on a remote server, learn how HTML and CSS direct the client computer to render a page, and experiment with JavaScript™ programming language to provide dynamic content. The focus of the unit is on the protocols that allow the Internet to function securely as it delivers social media and eCommerce content. Students work briefly in each of several web languages to understand how the languages work together to deliver this content. The history and workings of the Internet are explored, and issues of security, privacy, and democracy are considered. Practical cybersecurity hygiene is included. Career paths in cybersecurity, web development, and information technology are highlighted.

2.1 The Internet and the Web
2.2 Shopping and Social on the Web
2.3 Security and Cryptography

Unit 3: Raining Reigning Data

The goal of Unit 3 is for students to see the availability of large-scale data collection and analysis in every area they can imagine. Students examine very large data sets tied to themselves as well as to areas of work and society. They learn a variety of data visualization techniques and work to recognize opportunities to apply algorithmic thinking and automation when considering questions that have answers embedded in data. The complexity of the data sets, visualizations, and analysis increases in the second lesson of the unit, challenging students to generalize concepts developed in the first lesson.

3.1 Visualizing Data
3.2 Discovering Knowledge from Data

Unit 4: Intelligent Behavior

In Unit 4 the emergence of intelligent behavior is explored from two distinct approaches: from human crowd sourcing of data and from separate algorithmic agents working in parallel. The goal is to galvanize the connections among computing concepts and between computing and society. The exponential advancement of electronics, low on the ladder of abstraction, is connected to advancements at the highest levels on the ladder of abstraction, where artificial intelligence and simulation and modeling are impacting all fields. In the concluding lesson, students identify problems and questions that can be addressed with computer simulation, incorporating agent-based modeling. Students are challenged to explore the assumptions and parameters built into several simulations and to attach meaning to the results. Having explored a few applications of intelligent behavior emerging from algorithmic components, students reflect on the current and future state of artificial intelligence.

4.1 Moore's Law and Modeling
4.2 Intelligent Agents