AP Computer Science Principles
for Blue Valley Schools
Approved by College Board

Course Objectives

This course is based directly off of the College Board AP Computer Science Principles Framework. The main course objectives are summarized below in the six computational thinking practices and seven big ideas for the course.

Computational Thinking Practices:

The six computational thinking practices represent important aspects of the work that computer scientists engage in, and are denoted here by P1 through P6:

- P1: Connecting Computing
    - Identify impacts of computing.
    - Describe connections between people and computing.
    - Explain connections between computing concepts.
- P2: Creating Computational Artifacts
    - Create an artifact with a practical, personal, or societal intent.
    - Select appropriate techniques to develop a computational artifact.
    - Use appropriate algorithmic and information management principles.
- P3: Abstracting
    - Explain how data, information, or knowledge is represented for computational use.
    - Explain how abstractions are used in computation or modeling.
    - Identify abstractions.
    - Describe modeling in a computational context.
- P4: Analyzing Problems and Artifacts
    - Evaluate a proposed solution to a problem.
    - Locate and correct errors.
    - Explain how an artifact functions.
    - Justify appropriateness and correctness of a solution, model, or artifact.
- P5: Communicating
    - Explain the meaning of a result in context.
    - Describe computation with accurate and precise language, notations, or visualizations.
    - Summarize the purpose of a computational artifact.
- P6: Collaborating
    - Collaborate with another student in solving a computational problem.
    - Collaborate with another student in producing an artifact.
    - Share the workload by providing individual contributions to an overall collaborative effort.
    - Foster a constructive, collaborative climate by resolving conflicts and facilitating the contributions of a partner or team member.
    - Exchange knowledge and feedback with a partner or team member.
    - Review and revise their work as needed to create a high-quality artifact.

Big Ideas:

The seven big ideas of the course encompass foundational ideas of the field of computer science, and are denoted here by B1 through B7:

- B1: Creativity
    - How can a creative development process affect the creation of computational artifacts?
    - How can computing and the use of computational tools foster creative expression?
    - How can computing extend traditional forms of human expression and experience?
- B2: Abstraction
    - How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on a computer?
    - How does abstraction help us in writing programs, creating computational artifacts, and solving problems?
    - How can computational models and simulations help generate new understanding and knowledge?
- B3: Data and Information
    - How can computation be employed to help people process data and information to gain insight and knowledge?
    - How can computation be employed to facilitate exploration and discovery when working with data?
    - What considerations and tradeoffs arise in the computational manipulation of data?
    - What opportunities do large data sets provide for solving problems and creating knowledge?
- B4: Algorithms
    - How are algorithms implemented and executed on computers and computational devices?
    - Why are some languages better than others when used to implement algorithms?
    - What kinds of problems are easy, what kinds are difficult, and what kinds are
    - impossible to solve algorithmically?
    - How are algorithms evaluated?
- B5: Programming
    - How are programs developed to help people, organizations, or society solve problems?
    - How are programs used for creative expression, to satisfy personal curiosity, or to create new knowledge?
    - How do computer programs implement algorithms?
    - How does abstraction make the development of computer programs possible?
    - How do people develop and test computer programs?
    - Which mathematical and logical concepts are fundamental to computer programming?

- B6: The Internet
    - What is the Internet? How is it built? How does it function?
    - What aspects of the Internet's design and development have helped it scale and flourish?
    - How is cybersecurity impacting the ever-increasing number of Internet users?
- B7: Global Impact
    - How does computing enhance human communication, interaction, and cognition?
    - How does computing enable innovation?
    - What are some potential beneficial and harmful effects of computing?

The AP Performance Tasks:

The through course assessment is a set of performance tasks designed to gather evidence of student proficiency in the learning objectives. The AP Performance Tasks (PTs) are in-class assessments, administered by the teacher, that allow students to exemplify their learning through authentic, "real-world" creations.

The two performance tasks as defined by College Board are:

1. Explore - Implications of Computing Inventions
    - Students explore the impacts of computing on social, economic, and cultural areas of our lives.
2. Create - Applications from Ideas
    - Students create computational artifacts through the design and development of programs.

The AP Exam:

This course will prepare students for the multiple choice AP Computer Science Principles examination.

Note: Together, Software Development and Game Design (.5 credit) and AP Computer Science Principles (.5 credit) fulfill College Board's year-long (1.0 credit) curriculum for AP Computer Science Principles. All of the fundamentals and the required project are started in Software Development and Game Design and then continued into AP Computer Science Principles.

Course Outline

College Board Units 1 – 4: Content and Skills are addressed in Software Development and Game Design

Unit 5: The Internet (6 weeks)

Content	Skills
The internet, design and impact Internet hardware: networks, sRouters, abstractions in internet connections Internet addresses: IP, DNS, scaling Collaborative development and oversight	Demonstrate an understanding of how design affects: ● the reliability of network communication, ● online culture, ● the security of data, ● personal privacy, ● access to information, Discuss issues related to digital divide and intellectual property.
Routing: algorithms, redundancy, reliability Packets and Protocols: TCP/IP, HTTP Cybersecurity: problems, trust model, digital certificates	Analyze how protocols and algorithms are used in the internet, and the importance of cybersecurity.
Portfolio Problem	Explore the impact of internet on society. Choose an innovation that was enabled by the Internet and explore the positive and negative impacts of their innovation on society, economy, and culture Develop a computational artifact that illustrates, represents, or explains the innovation's purpose, its function, or its effect, and embed this artifact in a personal portfolio website.

Unit 6: Data (5 weeks)

Content	Skills
Impact of wealth of data	Analyze how the modern wealth of data collection has impacted society in positive and negative ways
Interpreting data: metadata, limitations on valid conclusions Visualizing data: produce and compare visualizations	Explore using computational tools to store massive amounts of data, organize and visualize data Find patterns in data Draw conclusions from data.
Collecting data	Identify Ways to store data Structure data sets for analysis Identify trade-offs between data storage schemas
Programmatically manipulating data APIs	Use computational tools and algorithms to search, sort and comparing algorithms Access data Use computational tools to find patterns in data.
Impact of Data Collection and Analysis Privacy rights	Predict with data models Analyze societal impacts of data collection and analysis
Portfolio Problem	Work in teams to investigate a question of personal interest, and use public data to present a data driven insight to their peers. Develop visualizations to communicate findings, and embed visualizations in a portfolio websites.

Unit 7: Performance Tasks (4 weeks)

Content	Skills
Review for AP performance task	Review course content Practice the skills necessary to complete AP performance tasks
Explore problem	Conduct efficient research Understand requirements for ● Project scoping ● Effective time management ● Incremental development ● Documentation ● Debugging ● Collaborative program development
Performance task	Complete AP performance task (12 hours of class time)

Unit 8: Review for the AP Exam (1 week)

Content	Skills
Prepare for Practice Exam	Review course content What to expect on the exam
Practice AP Exam	Cumulative Final AP Review Multiple Choice Test with immediate feedback

Unit 9: Final Project (Remainder of semester)

This project allows students to think creatively about the applications of the concepts covered in the course, and create something of personal value.

Content	Skills
Brainstorm	Brainstorm ideas for a final project, Discuss project ideas with peers
Project Planning	Project scoping Incremental development Plan out project milestones
Software Design	Designing an application from scratch Evaluate and compare proposed solutions Utilizing open source software
Implementation and Collaborative program development	Create final product Test and Debug
Final Demo and Presentation	Communicate the value of the final product via product demo, infographic, video, or a computational artifact of your choosing