Gateway-App Creators
Grade 8

Curriculum Committee Members
Matt McClellan, Career and Technical Education Coordinator

Reviewed by Curriculum Advisory Committee on March 1, 2018

Reviewed by High School Business Education Teachers on May 8, 2018

Approved by the Board of Education on June 19, 2018

COURSE TITLE: PLTW Gateway – App Creators

GRADE LEVEL: 8th

CONTENT AREA: Career and Technical Education

Course Description

App Creators introduces students to the field of computer science and the concepts of computational thinking, through the creation of mobile apps. Students are challenged to be creative and innovative, as they collaboratively design and develop mobile solutions to engaging, authentic problems. Students experience the positive impact of the application of computer science to society as well as other disciplines, particularly biomedical science.
The unit provides students opportunities for self-expression. Teams identify a personal or community problem of interest to them that can be solved with a mobile app solution. The problem can address issues such as health and wellness, the environment, school culture, emergency preparedness, education, community service—the options are endless! *Taken from pltw.org.*

Course Rationale

Students learn and apply computational thinking and technical knowledge and skills to create mobile apps. Students also acquire and apply skills pertaining to the design process, problem solving, persistence, collaboration, and communication. *Taken from pltw.org.*

Course Scope and Sequence

Unit 1: Let's Create an App! 15 class periods (90 minutes)	Unit 2: Taking It to the Next Level 15 class periods (90 minutes)	Unit 3: The App Challenge 15 class periods (90 minutes)

Proposed Course Materials and Resources

- Computer Lab/tablets/Chromebooks
- Inventory list provided by Project Lead the Way (www.mypltw.org)

Unit Objectives

Unit 1

The students will be able to:

1. Solicit and integrate peer feedback as appropriate to develop or refine a program.
2. Compare different algorithms that may be used to solve the same problem in terms of their speed, clarity, and size (e.g., different algorithms solve the same problem, but one might be faster than the other).
3. Interpret the flow of execution of algorithms and predict their outcomes.
4. Design, develop, and present computational artifacts such as mobile applications that address social problems both independently and collaboratively.
5. Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches.
6. Create variables that represent different types of data and manipulate their values.
7. Decompose a problem into parts and create solutions for each part.
8. Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively.
9. Use a systematic process to identify the source of a problem within individual and connected devices (e.g., follow a troubleshooting flow diagram, make changes to software to see if hardware will work, restart device, check connections, swap in working components).
10. Explain how computer science fosters innovation and enhances nearly all careers and disciplines.
11. Summarize security risks associated with weak passwords, lack of encryption, insecure transactions, and persistence of data.

Unit 2

The students will be able to:

1. Solicit and integrate peer feedback as appropriate to develop or refine a program.
2. Compare different algorithms that may be used to solve the same problem in terms of their speed, clarity, and size (e.g., different algorithms solve the same problem, but one might be faster than the other).
3. Interpret the flow of execution of algorithms and predict their outcomes.
4. Design, develop, and present computational artifacts such as mobile applications that address social problems both independently and collaboratively.
5. Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches.
6. Create variables that represent different types of data and manipulate their values.
7. Decompose a problem into parts and create solutions for each part.
8. Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively.
9. Use a systematic process to identify the source of a problem within individual and connected devices (e.g., follow a troubleshooting flow diagram, make changes to software to see if hardware will work, restart device, check connections, swap in working components).

10. Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools (e.g., use an app or spreadsheet form to collect data, decide which data to use or ignore, and choose a visualization method.).
11. Summarize negative and positive impacts of using data and information to categorize people, predict behavior, and make recommendations based on those predictions (e.g., customizing search results or targeted advertising, based on previous browsing history, can save search time and limit options at the same time).
12. Explain how computer science fosters innovation and enhances nearly all careers and disciplines.
13. Provide examples of how computational artifacts and devices impact health and well-being, both positively and negatively.
14. Redesign a computational artifact to remove barriers to universal access (e.g., using captions on images, high contrast colors, and/or larger font sizes).
15. Summarize security risks associated with weak passwords, lack of encryption, insecure transactions, and persistence of data.

Unit 3

The students will be able to:

1. Solicit and integrate peer feedback as appropriate to develop or refine a program.
2. Interpret the flow of execution of algorithms and predict their outcomes.
3. Design, develop, and present computational artifacts such as mobile applications that address social problems both independently and collaboratively.
4. Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches.
5. Create variables that represent different types of data and manipulate their values.
6. Decompose a problem into parts and create solutions for each part.
7. Use an iterative design process (e.g., define the problem, generate ideas, build, test, and improve solutions) to solve problems, both independently and collaboratively.
8. Use a systematic process to identify the source of a problem within individual and connected devices (e.g., follow a troubleshooting flow diagram, make changes to software to see if hardware will work, restart device, check connections, swap in working components).
9. Explain the processes used to collect, transform, and analyze data to solve a problem using computational tools (e.g., use an app or spreadsheet form to collect data, decide which data to use or ignore, and choose a visualization method.).
10. Explain how computer science fosters innovation and enhances nearly all careers and disciplines.
11. Describe ethical issues that relate to computing devices and networks (e.g., equity of access, security, hacking, intellectual property, copyright, Creative Commons licensing, and plagiarism).
12. Redesign a computational artifact to remove barriers to universal access (e.g., using captions on images, high contrast colors, and/or larger font sizes).

Essential Terminology/Vocabulary

Unit 1: Algorithm, animation, app, app development, app graphical design, conditional logic, debugging, design pattern, event-driven programming, media, MIT App Inventor, Model-View-Controller (MVC), pair programming, variables, and user interface features.

Unit 2: Algorithms, device sensors, iterate, loops, mobile applications, persistent data, and user input.

Unit 3: Computational thinking skills, design process, modules, principles, program, user-centered design, and user interface.