# **1- Creative Development**

Content Area:	Technology
Course(s):	
Time Period:	Marking Period 1
Length:	8 blocks
Status:	Published

## **Course Description & Instructional Notes**

# **Course Description**

Exam Weighting: 10-13% (also covered in Summer Assignment)

Collaboration is crucial when developing computing innovations, because having multiple perspectives offers opportunities to improve the design of innovations. In this big idea, students work collaboratively to design and develop programs using an iterative development process. They identify the needs of all users by gathering input from people from different backgrounds and demographics. Once the program is developed, they test it to ensure it meets these needs. Effective collaboration can often differ from group work, because it requires equal participation and voice from all members of the group. Early in the school year, it may be helpful for teachers to establish practices and norms that facilitate a collaborative environment and provide students with time to practice working together.

#### **Prior Knowledge**

none

#### **Instructional Notes**

Building Computational Thinking Practices:

When designing a solution to a problem, programmers consider both the program itself and the way the user will interact with the program: the user interface. A well-designed user interface makes it easy for the user to understand what data are required as input for the program to complete its tasks. When creating diagrams of their programs, students will benefit from considering how they want their program to behave based on identified inputs. Planning ahead may help them determine what abstractions can be developed and can help identify logic errors early in development. When implementing program design, programmers often use documentation to explain the purpose of various code segments and describe how they function together in the program. Students' diagrams can be a good place to start writing their documentation. It may be more helpful to concentrate on documenting smaller code segments rather than trying to describe larger sections all at once.

#### Preparing for the AP Exam:

Students will be expected to design and implement a program of their choice for the Create performance task. While students select their own topic for this task, they are required to include certain elements, such as lists and procedures, in their program code. Providing students with exemplars may help them consider the types of programs that can be developed while still meeting this requirement. Students will need practice identifying and correcting errors to prepare for the AP Exam. One way to give students this practice is to provide them with prewritten program code to correct.

#### **Technology Integration**

Computer Science naturally integrates technology on a daily basis.

## **Enduring Understandings**

Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.

Developers create and innovate using an iterative design process that is user focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.

# **Essential Questions**

How has working collaboratively with other students improved an overall project? What are some ways you can collect additional feedback on your program to use for improvements? What are some ways you currently plan your work before starting a project? What apps or programs have you stopped using because you didn't like the design of how you interacted with it?

# **Student Learning Objectives**

Students will be able to...

- Explain how computing innovations are improved through collaboration.
- Explain how computing innovations are developed by groups of people.
- Demonstrate effective interpersonal skills during collaboration.
- Describe the purpose of a computing innovation.
- Explain how a program or code segment functions.
- Identify input(s) to a program.
- Identify output(s) produced by a program.
- Develop a program using a development process.
- Design a program and its user interface.
- Describe the purpose of a code segment or program by writing documentation.
- Acknowledge code segments used from other sources.
- For errors in an algorithm or program: Identify and correct the error.
- Identify inputs and corresponding expected outputs or behaviors that can be used to check the

### **Vocabulary & Learning Experiences**

# **Essential Academic Vocabulary**

function, bug, debugging, syntax, start function, abstraction, loop, initialization, functions, conditions, control structures, comments, growth mindset, logical error, syntax error, run-time error, overflow error

#### **Planned Learning Experiences**

#### Sharing and responding

Have students develop a list of three questions that they would like to use data to answer. Then, in small groups, ask each student to share one of their questions. The group will respond with feedback to improve the focus and direction of the question. Students should take turns sharing their questions until all questions have been considered. Finally, ask each group to come to a consensus on which three questions they will answer with data.

#### Diagramming

In small groups, have students play a board game for 10 minutes. As they play, ask them to record the actions (such as rolling the dice or moving their piece) and decisions made in a diagram or flowchart. Have students trade games with another group and play the game using the diagram for directions. Students should identify and correct where the diagram might not be accurate or have missing steps. See the Language and Logic of Computing: Algorithmic Thinking Teaching and Assessing Module in the Professional Learning section of AP Classroom for a more detailed lesson plan and video example

#### Resources

CodeHS Code.org MobileCSP Google Classroom AP Classroom

#### Assessments

#### **Formative Assessments**

Quizzes embedded in CodeHS Modules and Code Review

The Two Towers: In this program, students have Karel build two towers of tennis balls. Each tower should be 3 tennis balls high. In the end, Karel should end up on top of the second tower, facing East. Students need to write at least 3 functions in order to solve this problem. This activity requires students to design and create functions for repeated processes within their program. Students need to consider top-down design and decomposition through the following questions: How can you break this problem down into smaller problems? What is a subtask that Karel needs to do more than once in this problem?

#### **Summative Assessments**

Introduction to Programming Unit Assessment (Multiple Choice online assessment provided by CodeHS)

Practice PT: Create Your Own UltraKarel Image

Following the milestones and the pseudocode plan that students have laid out, students use pair-programming to write the code for their final project. They then test their code along the way to make sure they have solved each milestone. This activity allows students to develop something completely unique with their programming skills and implement a successful algorithm of their own design.

Students then reflect upon and answer the following questions:

1. Identify the programming language and purpose of your program.

2. Describe the incremental and iterative development process of your program. How did you divide the program into smaller tasks and make a plan to complete them all?

3. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated.

4. Identify an algorithm that is fundamental for your program to achieve its intended purpose and includes two or more additional algorithms.

5. Describe how each algorithm within your selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program.

6. Identify an abstraction you developed, and explain how your abstraction helped manage the complexity of your program.

#### **NJSLS Standards**

NJSLS Standards Copied and Pasted as well as linked.

**NJSLS Computer Science and Design Thinking** 

8.2.12.ED.1: Use research to design and create a product or system that addresses a problem and make

modifications based on input from potential consumers.

8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.

8.2.12.NT.2: Redesign an existing product to improve form or function.

# **Additional NJSLS Standards**

NJSLS Standards Copied and Pasted as well as linked.

## Interdisciplinary Connections

<u>NJSLS Career Readiness, Life Literacies, and Key Skills</u>

9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas

9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice

9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving

9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.

9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task

9.4.12.TL.3: Analyze the effectiveness of the process and quality of collaborative environments.

9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem

**NJSLS Companion Standards Grades 9-12** (Reading & Writing in Science & Technical Subjects)

# GENERAL CONSIDERATIONS FOR DIVERSE LEARNERS

English Language Learners	Students Receiving Special Education Services	Advanced Learners	
	- Small group/One to one	- Use of high level	
	- Additional time	academic vocabulary/texts	
	- Review of directions	- Problem-based learning	
	- Student restates information		
- Personal glossary	- Space for movement or breaks	- Pre assess to condense curriculum	
- Text-to-speech	- Extra visual and verbal cues and prompts	- Interest-based research	
- Extended time	- Preferential seating	- Authentic problem- solving	
- Simplified / verbal instructions	- Follow a routine/schedule		
- Frequent breaks	- Rest breaks	- Homogeneous grouping opportunities	
	- Verbal and visual cues regarding directions and staying on task	Knowledge and Skill Standards in Gifted	
WIDA Can Do Descriptors for Grade 9-	- Checklists	Education for All Teachers	
<u>12</u>	- Immediate feedback	Pre-K-Grade 12 Gifted	
WIDA Essential Actions Handbook		Programming Standards	
FABRIC Paradigm	Students receiving Special Education programming have specific goals and objectives, as well as accommodations	<u>Gifted Programming</u> <u>Glossary of Terms</u>	
Wall Township ESL Grading Protocol	and modifications outlined within their Individualized Education Plans (IEP) due to an identified disability and/or diagnosis. In addition to exposure to the general education curriculum, instruction is differentiated based	Students with 504 Plan	
*Use WIDA Can Do Descriptors in	upon the student's needs. The IEP acts as a supplemental curriculum guide inclusive of instructional strategies that support each learner.		
coordination with Student Language		Teachers are responsible for implementing	
Portraits (SLPs).	Considerations for Special Education Students 6-12	designated services and strategies identified on a student's 504 Plan.	
	National Center on Universal Design for Learning - About UDL		
	UDL Checklist		
	UDL Key Terms		

entiation Strategies	
Independent Research & Projects	Jigsaw
Multiple Intelligence Options	Think-Tac-Toe
Project-Based Learning	Cubing Activities
Varied Supplemental Activities	Exploration by Interest
Varied Journal Prompts	Flexible Grouping
Tiered Activities/Assignments	Goal-Setting with
Tiered Products	Students
Graphic Organizers	Homework Options
Choice of Activities	Open-Ended Activities
Mini-Workshops to Reteach or Extend	Varied Product Choices
Think-Pair-Share by readiness or interest	Stations/Centers
Use of Collaboration of Various Activities	Work Alone/Together
	entiation Strategies Independent Research & Projects Multiple Intelligence Options Project-Based Learning Varied Supplemental Activities Varied Journal Prompts Tiered Activities/Assignments Tiered Products Graphic Organizers Choice of Activities Mini-Workshops to Reteach or Extend Think-Pair-Share by readiness or interest Use of Collaboration of Various Activities