02 User Events

Content Area: **Math**
Course(s):
Time Period: **Semester**
Length: **7 weeks**
Status: **Published**

General Overview, Course Description or Course Philosophy

The course is designed to give students an introduction to the programming language python and develop abstraction skills. Students will have the opportunity to foster their ability to develop and apply computational problem-solving skills. This course utilizes graphical problems that allow for multiple solutions and provides visual cues while debugging. Students will design creative solutions to address real-world problems. Students will engage not only independently but will work collaboratively to navigate the dynamic digital landscape.

In this unit students will have the opportunity to learn how to create functions to simplify code, select segments of code to be executed depending on conditions as well as how to create a dynamic program using keyboard and mouse interactions.

OBJECTIVES, ESSENTIAL QUESTIONS, ENDURING UNDERSTANDINGS

Objectives

- Students will be able to organize a set of instructions into a function.
- Students will be able to use user interaction to alter components of a canvas.
- Students will be able to explain what a parameter does and why a function might need one?
- Students will be able to compare and contrast mouse interactions.
- Students will be able to determine if an algorithm will run inside conditionals.
- Students will be able to write a helper function to simplify a long function.
- Students will be able to compare and contrast if, if-elif-else, and multiple if's.
- Students will be able to use the keyboard to interact with a program.
- Students will be able to compare and contrast nested conditionals and compound conditionals.
- Students will be able to explain the similarities and differences between && and ||.
- Students will be able to generate and test code to create complex shapes.

Essential Questions

- How do functions contribute to abstraction?

- How do programmers choose the correct data structure to manage a program's complexity?
- How do modules manage the complexity of a program?
- How does a programmer decide between readability, implementation, and performance?
- Why do programmers consider security, dependability, and accessibility when designing and evolving a program.
- Why are multiple approaches need when debugging a program?
- What are the advantages and disadvantages of using nested conditions instead of compound conditionals?
- How does a programmer use interaction from the user to control the flow of programming?

Enduring Understanding

- Instructions, control structures, functions, and expressions are used to create a program.
- Methods can be world or class level.
- Parameters act as a placeholder to communicate values from one method to another.
- Parameters allow a method to be used in more than one situation.
- Programming breaks down complex tasks into multiple simple actions.
- Programming uses mathematical and logical concepts.
- There are trade-offs when representing information as digital data.
- Programming uses mathematical and logical concepts to determine which sequence of code is executed.
- Python utilizes both mouse and keyboard for user interaction with a program.
- Properties of objects are not only defined within Python but can be customized to create new unique properties.
- Foundations of coordinate geometry are used when comparing the locations of objects on a canvas.

CONTENT AREA STANDARDS

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

RELATED STANDARDS (Technology, 21st Century Life & Careers, ELA Companion Standards are Required)

MA.K-12.1	Make sense of problems and persevere in solving them.
MA.K-12.2	Reason abstractly and quantitatively.
PFL.9.1.K12.P.4	Demonstrate creativity and innovation.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
PFL.9.1.K12.P.5	Utilize critical thinking to make sense of problems and persevere in solving them.
PFL.9.1.K12.P.6	Model integrity, ethical leadership and effective management.
PFL.9.1.K12.P.8	Use technology to enhance productivity increase collaboration and communicate effectively.
PFL.9.1.K12.P.9	Work productively in teams while using cultural/global competence.
LA.K-12.NJSLSA.R7	Integrate and evaluate content presented in diverse media and formats, including visually and quantitatively, as well as in words.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
LA.K-12.NJSLSA.W7	Conduct short as well as more sustained research projects, utilizing an inquiry-based research process, based on focused questions, demonstrating understanding of the subject under investigation.
LA.K-12.NJSLSA.SL1	Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
LA.K-12.NJSLSA.SL2	Integrate and evaluate information presented in diverse media and formats, including visually, quantitatively, and orally.
LA.K-12.NJSLSA.SL4	Present information, findings, and supporting evidence such that listeners can follow the line of reasoning and the organization, development, and style are appropriate to task, purpose, and audience.
LA.K-12.NJSLSA.SL5	Make strategic use of digital media and visual displays of data to express information and enhance understanding of presentations.

STUDENT LEARNING TARGETS

Declarative Knowledge

Students will understand that:

- Data can be stored and access using variables
- Variables can store information and objects
- Event handlers are used to activate sections of code when the user interacts with the program
- onMousePress and onMouseRelease are part of a larger group of event handlers
- Mouse interactions can be more complex than clicking

- Much like mouse interactions, key on a keyboard can be used to interact with a canvas.
- Properties are attributes of an object and can be altered with functions
- Conditional logic is used to control what code is executed
- If statements only run if the conditional is true
- Only the first true statement in an if-elif-else statement executes
- Conditionals can not only be complex but nested inside each other
- Functions are used to perform repetitive tasks
- Helper functions are an abstraction tool to simplify code
- Custom shapes are objects that specific methods can be applied to

Procedural Knowledge

Students will be able to:

- Create variables and use variables to access objects and information
- Create a function to group repetitive tasks
- Create helper functions to simply code
- Use functions to alter properties of shapes
- Use event handlers to create a responsive program
- Use if statements and conditionals to create more complex programs
- Display custom shapes on a canvas
- Manipulate custom shapes using methods
- Use keyboard interactions to alter the appearance of a canvas
- Alter the appearance of an object on the canvas using an if-elif-else statement

EVIDENCE OF LEARNING

Formative Assessments

- Checkpoints in each section
- Guided Practice
- Independent Practice
- Checklists
- Class Discussion
- Exit Tickets
- Rubrics
- Teacher Observation

- Exit/Entrance Tickets

Summative Assessments

- End of Unit Assessment
- End of Unit Creative Task

RESOURCES (Instructional, Supplemental, Intervention Materials)

Example Creative Tasks

INTERDISCIPLINARY CONNECTIONS

Interdisciplinary connections are frequently addressed through examples and practice problems whereby creating solutions that draw from cultures around the world, athletics, mathematics and geography. Examples can be found in topic specific examples, practice exercises, guided projects and digital resources.

ACCOMMODATIONS & MODIFICATIONS FOR SUBGROUPS

See link to Accommodations & Modifications document in course folder.