

## 1. Course Description

In the Bytes and Beats course, students will learn the fundamentals of programming with MATLAB while making their own music. They will use MATLAB, the programming language used by scientists and engineers, to turn if-statements, for-loops, and functions into rhythms, melodies, and harmonies.

Working with friends, students will compose and visualize their own music and invent their own symphonic creations. At the end of the course, students can take home their creations to continue fine-tuning their musical inventions.

No prior knowledge of programming or working with electronics is required.

Computer programming and musical concepts will be explored through various discussions, activities, lessons, and projects.

## 2. Learning Goals

At the end of each day, the students will be able to:

<b>Day 1</b>
<ul style="list-style-type: none"><li>- Explain the physical basis of sound</li><li>- Identify relationship between amplitude, frequency and shape of a sound wave to its volume, pitch and timbre</li><li>- Explain what programming is</li><li>- Open MATLAB and navigate the interface</li><li>- Execute commands in the Command Window and Live Scripts</li><li>- Create scalar and vector variables and perform basic operations.</li><li>- Identify string, logical and numeric data types and their sizes</li><li>- Index into vectors to extract values</li><li>- Parameterize their code to make it more flexible</li></ul>
<b>Day 2</b>
<ul style="list-style-type: none"><li>- Save variables to a MAT file and load them back</li><li>- Comfortably use vectors and variables</li><li>- Use the basic function-calling syntax</li><li>- Use pure sine tones to create a musical composition</li><li>- Use 'for' loop to write shorter and more efficient code</li></ul>
<b>Day 3</b>
<ul style="list-style-type: none"><li>- Refactor their code for better structure</li><li>- Create parameters in scripts</li><li>- Create a function</li><li>- Use logical operations</li><li>- Develop intuitive understanding of an interactive program's control flow</li><li>- Independently write conditional statements using logical operations</li></ul>
<b>Day 4</b>

<ul style="list-style-type: none"> <li>- Import and edit sound files in MATLAB</li> <li>- Understand that a sound signal is represented as a vector of numbers in MATLAB</li> <li>- Concatenate vectors of sound signals to combine them</li> <li>- Add two vectors of sound signals to mix them</li> <li>- Use addition and concatenation in a loop to create melodies</li> </ul>
<b>Day 5</b>
<ul style="list-style-type: none"> <li>- Apply all the concepts learnt through the course for personal projects</li> </ul>

### 3. Educational Standards

#### NGSS:

MS-ETS1-2. Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

MS-ETS1-3. Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

#### ISTE:

1. Creativity and Innovation:

- a. Apply existing knowledge to generate innovative ideas, products, or processes.
- b. Create original works as a means of personal or group expression.

4. Critical thinking, problem solving, and decision making:

- a. Identify and define authentic problems and significant questions for investigation.
- b. Plan and manage activities to develop a solution or complete a project.
- c. Collect and analyze data to identify solutions and/or make informed decisions.
- d. Use multiple processes and diverse perspectives to explore alternative solutions.

#### CSTA:

##### *Computational Thinking*

1. Use the basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing, evaluation).

##### *Computing Practice & Programming*

5. Implement problem solutions using a programming language, including: looping, behavior, conditional statements, logic, expressions, variables, and functions.

8. Demonstrate dispositions amenable to open-ended problem solving and programming (e.g., comfort with complexity, persistence, brainstorming, adaptability, patience, propensity to tinker, creativity, accepting challenge).