THE EDUCATION FUND

FOR EXCELLENCE IN MIAMI-DADE PUBLIC SCHOOLS

2020-2021

Ideas With IMPACT

idea packet
sponsored by:

FPL

Meet KIBO: Digital Literacy in the Primary Grades

MEET KIBO: CODING IN PRIMARY GRADES

By

Marcelle Farley

Table of Content

Robotics in Early Childhood

Why Robotics for Young Children Today?

Early childhood is a wonderful time to spark kids' interest in coding, robotics, and engineering. Young children are curious about the world around them, and today that world includes technology.

We are surrounded by technology. Smart phones, smart faucets, automatic doors. But in the early years, children learn very little about how these things work.

Integrating robotics into early childhood education helps close gender- and SES-based gaps in achievement in STEAM fields. For children in PreK-2, robotics can introduce STEM concepts to them in a playful way, thereby encouraging them to be excited about further exploration in Science, Technology, Engineering and Math!

Coding as a Playground

Research shows that young children can learn programming and engineering at a very early age. This is possible when children are given tools that are developmentally appropriate, that encourage open-ended play and that allow the integration of technical skills with expressive arts, math, literacy and cultural explorations.

Young kids learn by doing. They learn best by playing with physical objects: by making things, testing things. To learn programming and engineering, they need materials designed in the spirit of traditional learning manipulatives in early childhood (physical, as opposed to on-screen). Children engage in playful learning, cultivate their curiosity for the technological world, explore problem solving, and understand concepts such as sequencing, cause-and-effect, programming, sensors and motors.

https://kinderlabrobotics.com/robotics-in-early-childhood/

Six Key Benefits of teaching Robotics in Early Childhood Classrooms

1. Coding Teaches the Literacy of the 21st Century

We don't teach children to write because we want them all to be novelists or journalists; we teach them to write so they can express themselves. In the same way, teaching children to code gives them fluency in a new set of tools for self-expression.

2. Coding Develops Computational Thinking Skills

For example, when children use code to create algorithms (a series of ordered steps to solve a problem) they develop their sequencing ability: a foundational skill for reading and mathematics.

3. Technology Becomes the Playground

The best technology experiences for children are technology playgrounds, filled with creativity, exploration, and social engagement.

4. Robotics Makes Coding Tangible and Concrete… and Screen-Free!

Educational theorists have long recognized that young children think and learn best when moving, playing, building, and engaging with concrete objects.

5. Using Technology Breaks Down Engineering Stereotypes

Young children are naturally curious about the world around them. Today, that world includes technology. As adults – whether educators or parents – we have a responsibility to ensure that children's exploration of technology is age-appropriate, safe, and creative.

6. The Engineering Design Process Develops Grit and Perseverance

This process encourages children to identify a problem, imagine and plan a solution, build and test their creation, and share their work with peers.

ASK

Before engineers can start working on any project they have to ask what problem are they trying to solve.

What problem am I trying to solve? What do I want my robot to do?

Ask: What can I create to solve this problem. Draw a picture of what you can create or how your robot might look.

PLAN

Before building and programming a robot, engineers need to have a PLAN.
Which materials will you use to create our invention? Draw and label materials.

Draw a picture to show where all the parts will go.

CREATE

Now That you have a plan you can create your invention. Put all your materials together and Draw a picture of your creation.

Write a sentence that tells about your creation and what it does.

____

____

Draw a picture of your first program: Don't worry if the program doesn't work.

TEST & IMPROVE

K

Before the job is done, engineers need to **TEST & IMPROVE** their work. Use this checklist to see how your robot is coming along!

Is our robot sturdy?			
Do we have all of the sensors we need?			
Are all robot parts attached correctly?			
Does our robot look the way we want it to?			
Does our robot have all the motors it needs?			
Did we scan all of our blocks correctly?			
Did our robot do what we wanted it to?			

Now it's time to improve and fix your "bugs"! Don't be afraid to make changes. Make a tally mark in the box every time you change your robot or your program.

When making projects and building creations, engineers follow a series of steps called the "Engineering Design Process."

It has just 6 steps: ASK, IMAGINE, PLAN, CREATE, TEST & IMPROVE, and SHARE.

The Engineering Design Process is a cycle — there's no single starting point or ending point.

1. Which block goes first in every program?

2. Which block goes last in every program?

3. Which block tells KIBO to go straight ahead?

4. Which block tells KIBO to go in reverse?

5. Which block tells KIBO to stop and listen?

Some steps to follow to organize your students for robotics are:

• Research curriculum and resources and decide what you want to teach and how to integrate robotics in your required curriculum.

• Student teams for classroom purposes should be set up in groups of 2-4 students per robot.

• Define the role of each member of the team.

For instance:

(1) engineer (builder);

(2) software specialist (programmer);

(3) information specialist (gets the information and materials for the team to move forward or communicates the directions);

(4) project manager (keeps the team on task and facilitates problem solving).

• Identify what technology and materials are required. Such items as one computer per robot, one robot per team, ample space and storage, and backup components in case of breakdown should be considered. Also, make sure you have adequate system backup by using a separate disk or memory stick for each team.

• Write a budget and get funding. Consider robots, programming software, materials, etc. when planning.

• At the end of each lesson, share common problems and solutions, collect any written worksheets, provide test(s), if available, and move on to next lesson.

• At the end of instructional period, clean up Robot parts and return them to sorting trays, save all programs for each group, and assign any homework questions.

• You may decide to assign projects for a 2-3 week process. Be sure to provide a rubric of what is expected and plenty of time to complete. At the end of the project time, have each group present their solutions and how they arrived at them.

Team Activity Sheet

Project Title: ____

Name of team: ____

Team Members:

Engineer: ____

Programmer: ____

Information Specialist: ____

Project manager: ____

What materials did you use?

____

____

____

____

What problems did you encounter?

____

____

____

____

____

____

____

____

What were your solutions?

____

____

____

____

____

____

Engineer

- Responsible for building missions
- Design robot
- Draw plans

Programmer

- Write code for robot
- Test code on robot
- Explain code to group

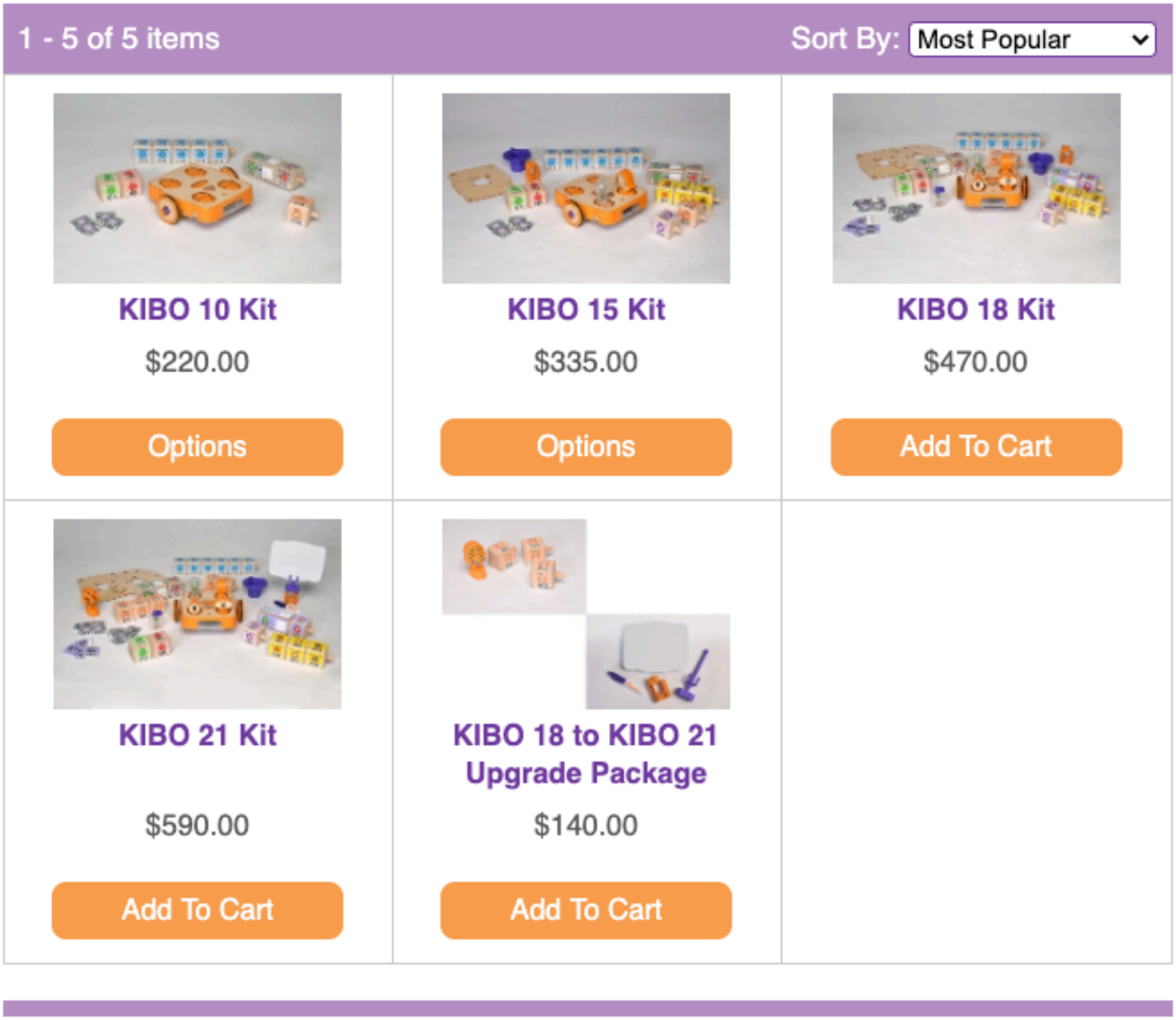
Information Specialist

- Research ideas for robot
- communicate ideas
- Check Plan for accuracy

Project Manager

- Keeps team on task
- Ensure all rules are being followed
- Collect materials

Cost of Kibo Kits



Resources

https://kinderlabrobotics.com/

mfarley@dadeschools.net