Delmock Technologies Inc. – Support for Makerspace Initiative Pilot Project
House Bill 1455 -- Senate Bill 0826

Challenge:
Makerspace Initiative Pilot Program in the Maryland Technology Development Corporation to encourage the establishment and expansion of makerspaces throughout the State; authorizing the Corporation to award certain financial assistance to local governments, certain designees of local governments, and certain nonprofit entities for the establishment of makerspaces in the State

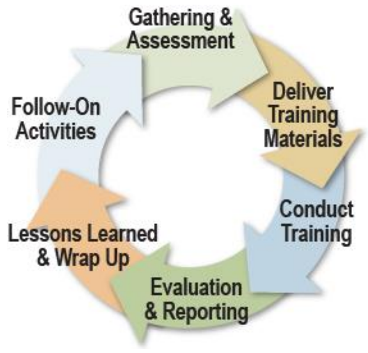
20 years later, Senator Barbara Mikulski vision for Maryland is still being realized "we're wired and inspired". Maryland is home to the nation's top security agencies – the U.S. headquarters for cybersecurity. Two decades later we have an opportunity to ensure that vision continues and includes Maryland high school students that lack the resources and industry engagement to be part of the wired and inspired journey of advanced technologies.

Solution:
Helping to introduce a new generation of Maryland high school students to the world of Advanced Technology with the hope it will pique an interest in cybersecurity and robotics. These high school students if provided the right education and skills training can begin to fill the advanced technology—based worker shortages within their perspective communities of Prince George county and the state.

That right choice will include learning the fundamental of the Python programming language. Python is easy to use, powerful, and versatile, making it a great choice for the high school students. Python's readability makes it a great first programming language. It will allow the participants to quickly think like a programmer.

Guide:
As a member of the Advanced Technology Center (ATC) project, Delmock Technologies, Inc (DTI) is in the unique position to help increase workforce development by launching a program at Gwynn Park and Surrattsville High Schools, in Prince Georges County Maryland. A curriculum designed to prepare the next generation of cybersecurity and robotics professionals for the emerging autonomous vehicle technology workforce.

Plan:
Goals. A program which stimulates a career interest in technology: acquisition of technology-oriented work opportunities upon high school graduation; pursuing post-high school technology certifications, college credits and/or degrees; and pipeline of talent for the local Maryland-based IT companies.
Participants' Requirements. 11th grader with a C or above average; willing and able to meet twice a week for duration of 2 hours; for a total of four academic quarters.
Gathering & Assessment. Gathering of participatory data to facilitate the customization of the training lessons to meet the goals of the program. A questionnaire will be used to assess the students' "as is" technical level. Once the appropriate data is gathered, an integrated training solution is designed and implemented. Training solutions will align with the ATC workforce needs while leveraging industry best-practices creating a training baseline for all curriculum modules. DTI determines what knowledge needs to be transferred to improve the skillsets of the individual helping them move from the current technical abilities to meet the goals of the program.



Design Right Training Materials. Each lesson is aimed to challenge the student (interactive, hands-on, field-trips, etc.). Messaging designed to engage Generation Z. *[see sample curriculum overview]*. DTI's training materials is based on the most popular language "Python".

Conduct Training. "How To" documentation; Instructor-led training; and Online labs.

Evaluation & Reporting. Student surveys are conducted at the end of each quarter to determine and assess level of knowledge transfer and results are reported to the appropriate stakeholders.

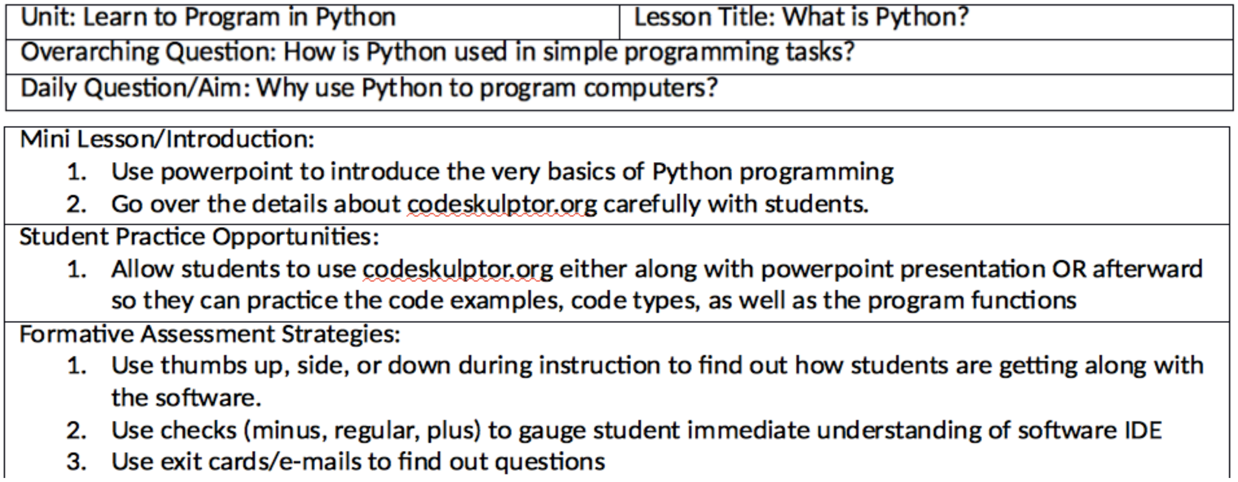
Lessons Learned & Wrap-up. DTI considers success as an iterative process, one that continues to measure progress, share lessons learned and provide opportunities for knowledge transfer.

Follow-On Activities. Customized process designed to utilize the efficient elements of the existing training processes (follow-on activities) while infusing them with key processes and lessons-learned.

Sample Curriculum Overview

1. Curriculum will employ instructor to student ratio of 14:1, and incorporate the planned interaction of pupils with instructional content, materials, resources, and processes for evaluating the attainment of educational objectives. Lessons will be aligned to local and national educational standards. Assessment efforts will, in the best circumstances, reflect educative or authentic practices to engage students. Our curriculum will prepare and inspire students to think and explore.
2. Curriculum is built upon Python, one of the most popular and easiest introductions into coding and program in the industry today. Python is built upon project-based learning which gets high school students engaged. We understand from professional in the field that the key to learning Python is working with small chunks of information at a time that is organized into a highly visual format.
3. Curriculum will teach students about Python functions and modules, tips and tricks, coding, and introduce them to the field of penetration testing.
4. Curriculum will employ CodeSkulptor which was created in 2012, and used as a tool for teaching Python programming, especially to beginners. A driving goal is to be very easy to use. CodeSkulptor uses two specializations that currently run on Coursera, "Fundamentals of Computing" and "An Introduction to Scripting in Python." Some of its main advantages for teaching are:
   - students do not need to install any software;
   - students will all have the same Python version and the same editor; and
   - students can access the same programming environment and code files from any computer.

Sample Lesson Plan



Unit: Learn to Program in Python	Lesson Title: What is Python?
Overarching Question: How is Python used in simple programming tasks?	
Daily Question/Aim: Why use Python to program computers?	

Mini Lesson/Introduction:

1. Use powerpoint to introduce the very basics of Python programming
2. Go over the details about codeskulptor.org carefully with students.

Student Practice Opportunities:

1. Allow students to use codeskulptor.org either along with powerpoint presentation OR afterward so they can practice the code examples, code types, as well as the program functions

Formative Assessment Strategies:

1. Use thumbs up, side, or down during instruction to find out how students are getting along with the software.
2. Use checks (minus, regular, plus) to gauge student immediate understanding of software IDE
3. Use exit cards/e-mails to find out questions