

**Intro to Computer Programming  
Curriculum Map**

	<b>Themes &amp; Essential Questions</b>	<b>Standards-based Essential Skills/Concepts to Targeted &amp; Instructional Strategies</b>	<b>Formative/Summative Assessments</b>
<b>1<sup>st</sup> Quarter</b>	<ol style="list-style-type: none"> <li>1. Background of programming What is the difference between explicit and implicit directions?</li> <li>2. Introduction to Visual Basic/Learning the language What is the purpose a programming language? Why are there some many different ones?</li> <li>3. Data and Variables What information is needed to be able to properly perform a task?</li> <li>4. Control structures What the different control structures available and what do they do for you?</li> </ol>	<p>The course follows the “Steps of the Engineering Design Process” as listed in the <i>Science and Technology/Engineering High School Standard</i>:</p> <ol style="list-style-type: none"> <li>1. Identify the need or problem</li> <li>2. Research the need or problem Examine current state of the issue and current solutions Explore other options via the Internet, library, interviews, etc.</li> <li>3. Develop possible solution(s) Brainstorm possible solutions Draw on mathematics and science Articulate the possible solutions in two and three dimensions Refine the possible solutions</li> <li>4. Select the best possible solution(s) Determine which solution(s) best meet(s) the original requirements</li> <li>5. Construct one or more prototypes and/or models Model the selected solution(s) in two and three dimensions</li> <li>6. Test and evaluate the solution(s) Does it work? Does it meet the original design constraints?</li> <li>7. Communicate the solution(s) Make an engineering presentation that includes a discussion of how the solution(s) best meet(s) the needs of the initial problem, opportunity, or need Discuss societal impact and tradeoffs of the solution(s)</li> <li>8. Redesign Modify the solution(s) based on information gathered during the tests and presentation</li> </ol>	<ol style="list-style-type: none"> <li>1. Students prepare directions for simple tasks and have other students try to complete the task by following the given directions.</li> <li>2. Students research a programming language and do a presentation on it explaining the language’s history, specific purpose, and current area of use. Students are assigned a Visual Basic toolbox object and prepare a presentation to teach the rest of the class about it.</li> <li>3. Students create a short program designed to perform a simple task.</li> <li>4. Students create a program that must make decisions and must repeat to be able to perform the assigned task.</li> </ol> <p><u>Quarterly Assessment:</u> Students must be able to identify the purpose of various parts of a programming language. They must be able to recognize proper language syntax and data usage. They must be able to identify the proper control structure type needed to perform a specified task and demonstrate its proper usage.</p>

	Themes & Essential Questions	Standards-based Essential Skills/Concepts to be Targeted & Instructional Strategies	Formative/Summative Assessments
2 <sup>nd</sup> Quarter	<p>1. Functions What is a function and what does it do for you? What are the major functions included in Visual Basic and what is their purpose?</p> <p>2. User Interface What are the elements of good user interface design?</p> <p>3. Debugging How do you know when your program is working properly? How do you fix it if it isn't?</p> <p>4. Program Documentation What documentation is needed and what is its purpose?</p>		<p>1. Functions are introduced as a way to provide shortcuts for concepts used repeatedly. Students explore using built-in functions as well as creating their own functions.</p> <p>2. The importance of good user interface is introduced. Students research user interface and HCI topics and do presentations. They redesign some of their earlier programs to reflect what they learned about good user interface design.</p> <p>3. Quality control in the form of program testing is introduced. Students must prepare Q/A guidelines that are able to determine if a program is performing correctly. Debugging concepts and techniques are introduced and students must take a program someone else has written and debug it and then show that it meets the program specification requirements.</p> <p><u>Quarterly Assessment:</u> Students must be able to complete a fully documented large project based on program requirements. The program must include correct usage of data, control structures, functions and good user interface design techniques. They student must be able to show that the completed project meets the requirements.</p> <p><u>Final Assessment:</u> They must be able create a properly documented, working short program based on specified requirements.</p>