WASHINGTON LEADERSHIP ACADEMY

Advanced Math II: complex functional relationships and modeling in computing

Bootstrap

+ computing creatively thriving mathematically

(A course adapted from and made in partnership with Bootstrap)

Advanced Math II is an implementation of the Bootstrap:2 curriculum (bootstrapworld.org), which goes deeper into programming. This course builds events and data structures on top of the foundation laid by Advanced Math I, and introduces a brand new programming language called Pyret, which blends the pure-algebra semantics that make Bootstrap successful with a Python-like syntax. The knowledge and skills learned in this course will allow students to build far more sophisticated programs than in the previous course. As part of the curriculum, students learn how the world-based event loop that drives their original, Advanced Math I video game works, and use this event loop to create animations using simple data types for their world. They then learn about data structures, and design a World structure for a sophisticated game of their own design. This course introduces students to complex functional relationships in math, event-driven programming, data structures, data modeling, encapsulation, and explores applied randomness in math and computer programming.

Common Core Alignment Advanced Math II is aligned to Common Core Standards for Mathematics, covering most Functional and Algebraic standards across the Common Core from Grade 7 through Algebra 2. This includes Mathematical Practice standards as well as Mathematical Content standards. Additionally, Advanced Math II is aligned to the CSTA Computer Science standards. This alignment makes it possible and flexible for teachers to integrate the course into the classroom. Because Advanced Math II is a continuation of Advanced Math I, and primarily a programming course, it is more closely aligned with the CSTA Computer Science standards in each module.

Core Competencies:

Computational Thinking Practices

- Analyze the effects of the developments in computing
- Design and implement creative solutions and artifacts
- Apply abstractions and models
- Analyze your own computational work and the work of others
- Communicate computational thought processes, procedures, and results to others
- Collaborate with peers on computing activities

Standards of Math Practice

- Make sense of problems and persevere in solving them
- Reason abstractly and quantitatively
- Construct viable arguments and critique the reasoning of others
- Model with mathematics
- Use appropriate tools strategically
- Attend to precision
- Look for and make use of structure
- Look for and express regularity in repeated reasoning

Modules Overview:

Module 1: Everything you learned in Advanced Math I
Module 2: Introduction to Data Structures
Module 3: Structures and Worlds
Module 4: Welcome to the World
Module 5: Key Events
Module 6: Asking the World
Module 7: Collision Detection
Module 8: Building Your World
Module 9: User Interaction and Complex next-world in Your Game
Module 10: Collision Detection and Completing Your Game
Module 11: Scoring, Levels, and Nested Structures

Module Assessments

A Common Timeframe for Modules and Interim Assessments The Advanced Math II curriculum will utilize performance-based assessments after each module to enhance and reinforce learning, while giving students a project in which to apply the algebra and programming skills learned in each module. Group-based performance tasks will also be given at regular intervals to reinforce the teamwork and communication skills embedded into the curriculum. These tasks will align themselves to the PARCC Performance Level Descriptors.

Instructional Model and Implementation Strategies for Advanced Math II

Students will take Advanced Math II every other day for 90-minutes for the duration of one interim. The course is intended to thread together Algebraic concepts into computing. Adding from skills acquired in Advanced Math I, students further develop functional programming skills that allow them to add layers of data structures using a series of mathematically driven arguments. This model allows students to further understand math concepts and how they intersect with computing. These outcomes are met using the following instructional strategies:

Accountable video-based Instruction: videos used to introduce or review a specific scientific concept that includes practice problems and scenarios within the video ensuring that students attention and time spent. Student usage data is tracked as well as their achievement data on video-associated assessments.

Design Nodes: Students engage in a stepwise design process to achieve a large outcome. Students work through smaller outcomes (nodes) that when combined or sequenced together make the large goal of the process occur.

Skills-based Workshops: Students work through a self-paced skills workshop designed to lead students to acquire a specific skill. Workshops typically use a gamified or challenge-styled approach to building deepening levels of complexity around a skill.

Engineering and Design Thinking Challenges: students are provided with a challenge to program or code a solution, test prototypes, and design new models around a computing outcome. Students work independently and collaboratively to reach specific outcomes.

Standards Coverage

Interim Session	Date Range	Standards Covered
As Applies	**As Applies** (47 days)	***CCSS*** N-Q, 5.OA.1-2, 6.NS.5-8, 7-EE.3-4, 8.G.6-8, 8.F.1-3, 8.F.4, 8.F.5, A-SSE.1-2, A-CED.1-4, 8.G.6-8, F-IF.1-3, F-IF.4-6, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***SMPs*** MP.1-8s ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7

Module #1: Everything You Learned in Advanced Math I

While learning a new programming language, students review the major concepts and material from Bootstrap:1, including Contracts, Expressions dealing with numbers, strings and images, Variable definitions, Function definitions, and the Design Recipe. Students define functions in Pyret based on problem statements.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 7-EE.3-4, 8.F.1-3, 8.F.5, A-SSE.1-2, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A3, L3:CPP:A4	**Bootstrap:2 Unit 1** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit1/index.html**	**Bootstrap PBA I and Performance Tasks**

Module 2: Introduction to Data Structures

Students discover the need for data structures by identifying real-world behaviors that require them (a falling parachute jumper) and they practice defining data structures. They make use of a complex, pre-defined data structure: Cake. Students will define variables bound to Cakes and write code that extracts each field from those Cakes. Students will also generalize their understanding of function constructors and accessors over data structures.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 7-EE.3-4, 8.F.1-3, 8.F.4, 8.F.5, A-SSE.1-2, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:7, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A3, L3:CPP:A4, L3:CT:B5, L3:CT:B6, L3:CT:B10	**Bootstrap:2 Unit 2** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit2/index.html**	**Bootstrap PBA 2 and Performance Tasks**

Module #3: Structures and Worlds

Students, having worked with pre-made data structures (Cakes) in the last lesson, generalize their understanding by defining more data structures of their own, accessing their fields, and writing functions that produce them. Students will define a new data structure: a party, and write functions that access fields of a Cake, and write functions that produce new Cakes.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 7-EE.3-4, 8.F.1-3, 8.F.4, 8.F.5, A-SSE.1-2, A-CED.1-4, 8.G.6-8, F-IF.1-3, F-IF.4-6, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:7, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 3** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit3/index.html**	**Bootstrap PBA 3 and Performance Tasks**

Module #4: Building Evidence- Based Arguments

Students begin this module by altering and adding fields to the Cake data structure. From there, they are introduced to event-based programming using Pyret's event-based programming model big-bang. Students will act out the event-based programming model, beginning with a simple world containing just one number, to understand how the model works. From there, they will codewalk through the next-world and draw-world functions of a simple videogame. They then modify these functions and experiment with the results, eventually leading to a point where they discover the need for World structures.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 7-EE.3-4, 8.F.1-3, 8.F.4, 8.F.5, A-SSE.1-2, 8.G.6-8, F-IF.1-3, F-IF.4-6, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 4** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit4/index.html**	**Bootstrap PBA 4 and Performance Tasks**

Module #5: Key Events

Students return to the subject of piecewise functions, beginning by codewalking through a piecewise function they wrote in Advanced Math I, identifying the differences in Pyret syntax. From there they define a key-event handler that modifies a data structure (World) when certain keys are pressed.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 7-EE.3-4, 8.F.1-3, 8.F.5, A-SSE.1-2, 8.G.6-8, F-IF.1-3, F-IF.4-6, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 5** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit5/index.html**	**Bootstrap PBA 5 and Performance Tasks**

Module #6: Asking the World

Students continue to combine their use of Ask and Data Structures, this time identifying ways in which the World structure of their Ninja Cat videogame might change without any user input. They will expand their next-world function from a linear function into a piecewise function to account for changes in the world structure. Students will learn about the random function, and use it to make game characters appear at different locations on the screen.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 6.NS.5-8, 7-EE.3-4, 8.F.1-3, 8.F.5, A-SSE.1-2, A-CED.1-4, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 6** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit6/index.html**	**Bootstrap PBA 6 and Performance Tasks**

Module #7: Collision Detection

Students return to the Pythagorean Theorem and distance formula they used in Advanced Math I, this time with data structures and the full next-world function. Students will write the distance function in Pyret, as well as a function to tell whether two characters have collided. They will use different Ask branches within their next-world function to identify collisions in their Ninja Cat games.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 6.NS.5-8, 7-EE.3-4, 8.G.6-8, 8.F.1-3, 8.F.5, A-SSE.1-2, A-CED.1-4, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 7** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit7/index.html**	**Bootstrap PBA 7 and Performance Tasks**

Module #8: Building Your World

After thinking about their own game World, students practice building, drawing, and animating it by defining their own World structure, and writing their own draw-world and next-world functions.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 7-EE.3-4, 8.F.1-3, 8.F.4, 8.F.5, A-SSE.1-2, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 8** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit8/index.html**	**Bootstrap PBA 8 and Performance Tasks**

Module #9: User Interaction and Complex next-world in Your Game

Using the same techniques as their Ninja Cat videogames, students will return to piecewise functions to write a function to make their own games respond to keypresses, and identify circumstances in which the functions in their games should behave differently. They will define these circumstances - and the desired behavior - in code, as different ask branches in their next-world functions, to program ways in which their own world structure will change without user input.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 6.NS.5-8, 7-EE.3-4, 8.F.1-3, 8.F.4, 8.F.5, A-SSE.1-2, A-CED.1-4, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 9** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit9/index.html**	**Bootstrap PBA 9 and Performance Tasks**

Module #10: Collision Detection and Completing Your Game

Using the distance and is-collision functions they wrote for their Ninja Cat games, students complete collision detection in their own videogames, and return to the lists of questions to ask their world from the previous module to complete their games. Students will use different Ask branches within their next-world functions to identify collisions in their games.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 6.NS.5-8, 7-EE.3-4, 8.G.6-8, 8.F.1-3, 8.F.5, A-SSE.1-2, A-CED.1-4, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Unit 10** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/unit10/index.html**	**Bootstrap PBA 10 and Performance Tasks**

Module #11: Scoring, Levels, and Nested Structures

Students will expand and add features to their games, including a scoring system and levels, which alters the appearance of their world by changing their draw-world function into a piecewise function to create different levels. Students will also use nested structures to add further complexity to their games.

# of Days	Standards	Course Resources	Assessments
2	***CCSS*** N-Q, 5.OA.1-2, 6.NS.5-8, 7-EE.3-4, 8.F.1-3, 8.F.5, A-SSE.1-2, A-CED.1-4, 8.G.6-8, F-IF.1-3, F.IF.7-9, F-BF.1-2, F-BF.3-4, F-LE.1-4, F-LE.5, F-TF.5 ***CSTA Standards*** L1:6:CT.1, L1:6:CPP.6, L2:CT:1, L2:CT:6, L2:CT:7, L2:CT:14, L2:CPP:4, L2:CPP:5, L2:CPP:9, L3:CT:A2, L3:CPP:A2, L3:CPP:A3, L3:CPP:A4, L3:CPP:A8, L3:CPP:A12, L3:CT:B5, L3:CT:B6, L3:CT:B10, L3:CPP:B2, L3:CPP:B7	**Bootstrap:2 Supplemental** **http://www.bootstrapworld.org/materials/spring2016/courses/bs2/units/Supplemental/index.html**	**Bootstrap PBA 11 and Performance Tasks**