



Title: Algebra 2 for the 21st Century

Length of Course: 1.0 unit (one year, two semesters or three trimesters equivalent)

Subject Area – Discipline: Mathematics (“c”) - Algebra 2

CTE Sector: Information and Communication Technologies

CTE Pathway: Software and Systems Development

Grade/Level(s):

Pre-Requisite(s):

Course Overview:

This course is designed for students who are passionate about applications of mathematics and have a desire for a career in computer programming. In this course, Algebra 2 standards are combined with the game, simulation, and software development concepts into an integrated secondary curriculum that meets both Algebra 2 course requirements and CTE standards. Students will research, analyze, and modify existing program code and develop their own program code that will integrate major Algebra 2 concepts in each of the six units; linear functions, quadratic functions, polynomial function, rational expressions and equations, exponential and logarithmic functions, and systems of equations. Through the unit programming projects, students understand and master the mathematics and programming code necessary in the development of games such as Pong and Angry Birds and the simulation of fractal images based on iterating rational functions. Students will also explore some ethical issues around the rapid development of technology and its impact on society.

Course Content:

[Unit 1 - Linear Pong](#)

This unit will provide students with both a review in Algebra 1 topics as well as an immediate "hook" for the students provided by delving into the world of programming games. Students will be exposed to immediate uses for the math they are asked to learn. Students receive instruction on operations with linear functions and their graphs, slope and rate of change, graphing of linear equations, and writing linear equations. Students practice these skills through practice problem sets.

The programming integrated development environment (IDE) is introduced to the students with a ball (circle) travelling horizontally across the screen, bouncing as it encounters the vertical edges. As the teacher questions the students about the possibilities of making the ball's path change, students will be led to investigate the idea

of slope in linear equations as they learn the fundamentals of programming. This is the starting point of where we can discuss how slope is used in the game of Pong. An understanding of slope is necessary to have the ball travel in a diagonal direction and have realistic bounces. Students take the initial horizontal ball program provided by the teacher and add functionality to create the game of Pong: give the ball the ability to travel using a non-horizontal slope and user interface.

Besides being a good review of slope for students, Pong is a perfect introductory project because it uses much of the basic structures of programming without being overly complicated. Students can see the uses and form of the structures as they relate to a tangible idea: the movement of a ball. Students will also see the relationship between the equations of lines and the path of the pong ball sprite. Students will also develop a users manual for the game.

Major Topics:

- ★ Review slope and rate of change
- ★ Graph linear equations
- ★ Write equations of lines
- ★ Understand, use, and interpret slope-intercept form of linear equation to describe the path of the ball in the game of Pong
- ★ Control the ball using the change in x and the change in y from the slope; change in y/ change in x
- ★ See the effect of horizontal slope in the context of the game
- ★ Use congruent angles to program realistic looking bounces
- ★ Translate the path of the ball into a linear graph on a coordinate plane Identify phases of software development life cycles
- ★ Work as a member of a development team
- ★ Use basic input/output structures to make the game interactive
- ★ Program control structures such as conditional and iteration
- ★ Debug software
- ★ Use media design and editing software

Unit 1 Assignments

Programming Task: Pong Students will be given a teacher generated program with a horizontal bouncing ball. This will provide the students a starting point for working with the language and mechanics of programming and the teacher with a point of reference to begin instruction in both programming and advanced algebra skills. This assignment will be completed in five parts to be checked by the instructor before students move on to the next step in the program until they have completed writing a functional "pong" game:

- 1) Paddles for two players
- 2) Detection of collision between ball and paddle (bounce), collision between ball and horizontal boundaries (bounce), and ball and vertical boundaries (score).
- 3) The ability to move the paddles
- 4) Score-keeping 5) Display of the score and termination of game once a determined

score has been reached.

User Manual Students develop a user manual to instruct a user how to play the game. This should include rules, starting the game, moving the paddles, and tips for winning. A good user manual will be approximately one page and include math vocabulary relevant to the unit (eg "When the paddle intersects with the ball, the ball will bounce away with an opposite slope.").

Weblog Students will start a blog that they will add to at least once per unit, showing connections between their learning of programming and its connection to their learning of mathematics. Prompt: "Describe how the slope of the pong ball sprite changed as it collided with the paddle and horizontal edge." (Approximately one page.) Students will read and respond to 3 peer blog entries.

Reading - Chapter 1 of *Blown to Bits: Digital Explosion* Write a 3 page paper reflecting how the perforation of information "digital explosion" is changing the way we live. In addition, students will participate in a panel discussion. This will help students understand the complexity of the issues of creating programs (including games) and their ethical usage.

Skill Practice sets from State Approved Algebra 2 Textbook - Since the ball models a linear equation, students will practice graphing various lines with different slopes and y intercepts and writing equations from graphs. When they realize the game with a slope of zero (horizontal) is boring, an understanding of these concepts will help them modify their program to be more exciting (eg, Should the ball have a large slope? small? negative? What does the y intercept mean in the game? How can we tell if the ball hits the paddle? What should happen to the slope of the ball if this happens?, etc)

[Unit 2 - Projectiles](#)

Upon completion of the basic Pong program, the teacher will assign students a slightly more complex program to build on their programming skills as well as provide an application of the Algebra 2 skills that will be taught during this unit. Students receive direct instruction and engage in guided practice for right triangle trigonometry, evaluating parametric equations, and major quadratic topics including evaluating, solving with a variety of methods, and finding minimum/maximum values and zeros. Students analyze the effects of coefficients in the standard form of quadratic function with respect to projectile motion meaning gravity, initial velocity, and initial height. Pair and share would be a useful instructional method, as well as cooperative learning groups. Students practice these skills through practice problem sets. Upon mastery of these skills, students with a learning partner will write a program that will utilize these skills to model parabolic projectile motion. students write reflections of their learning process and how variables are similar and different in algebra and programming in their weblog. They will be prompted to use the "I know, I want to know and I learned" strategy for their blog entry. The study of quadratics lends itself to creating an "Angry Birds" style game, whereby students will have to expand on both their knowledge of quadratics as well as

their programming skills to successfully create.

Major topics:

- ★ Introduction to properties of quadratic functions leading to investigations of parabolic functions
- ★ Utilize right triangle trigonometry to create parametric projectile motion equations to be used in writing a program
- ★ Calculate minimum/maximum values and zeros of quadratic functions and apply to writing a program
- ★ Use projectile motion equation within the programming key assignments
- ★ Obtain zeros of a function
- ★ Given initial velocity and height, create quadratic equations Use advanced math functions within programming key assignments
- ★ Debug larger programs in the context of the programming key assignments
- ★ Discover the relationship between the use of variables in mathematics and computer science
- ★ Review and reinforce programming concepts from Unit 1.

Unit 2 Assignments

In order to scaffold students' understanding of parabolic motion, we will break the task into two parts One isolating the problem in only the y direction, simplifying the problem to get them started. Once the students show understanding of the one dimensional motion of a projectile through their program, we will ask them to add an x component so the projectile moves through space and time. Finally, we will add obstacles to situation to simulate an "Angry Birds" game.

Programming Task 1 - Projectile Motion in 1-dimension students will use $y = -12gt^2 + v_0t + h_0$ to model the path of the projectile in the program. The students create a program to show an object thrown straight up in the air.

Programming Task 2 - Projectile Motion in 2-dimensions students will use $x = v_0t \cos$
 $y = -12gt^2 + v_0t + h_0$ to model the path of the projectile in the program. The students write a program to show an object being launched at an assigned angle and initial velocity.

Skill Practice sets from State Approved Algebra 2 Textbook - Students complete daily practice sets to reinforce the Algebra 2 skills on right triangle trigonometry, evaluating parametric equations, and major quadratic topics including evaluating, graphing, solving with a variety of methods, and finding minimum/maximum values and zero.

Weblog Daily Prompt: "Describe in detail one thing you learned, one thing you want to know more about, and one thing you have trouble with." (Approximately one page.)
Prompt: "How are variables used in Algebra similar to variables in programming? How are they different?" (Approximately one page.) Students will read and respond to 3 peer blog entries.

Unit 3 - Factoring

In this unit students build on their programming skills after learning about factoring polynomial expressions and solving polynomial equations with rational and complex roots. Students practice these skills through practice problem sets from the Algebra 2 textbook. Once students have mastered the skills of factoring and solving polynomial expressions, they will research and modify an existing program from the online Scratch library so that it will find the prime factorization of an integer. In order to ensure mastery of Algebra 2 and reinforce knowledge and skill of data and control structures students will complete a group programming project. In this project they will extend the modified program that do prime factorization to factor polynomials up to degree three. The programming assignments will require students to use conditional statements, iterative structures, and built-in math libraries. Students also explore and analyze the effects of the digital explosion in society in how information is affecting every aspect of daily life. They will explore the legal and ethical implications of computing on our society. Small group learning and panel discussions will help facilitate this learning.

Major Topics:

- ★ Major Topics: Review and analyze prime factorization of integers.
- ★ Factor polynomials including rational and complex roots
- ★ Create programs that use conditional and iterative control structures
- ★ Demonstrate the use of math functions built in the programming library to factor polynomials
- ★ Discuss and explore changes in technology and lifestyles and its impact on privacy

Unit 3 Assignments

Weblog Prompt "Describe the need to master the pencil and paper process before applying the programming process. Evaluate and analyze the benefits and obstacles of both methods." (Approximately 2 to 3 pages.) Students will read and respond to 3 peer blog entries.

Reading: Chapter 2 of *Blown to Bits: Your Life, Liberty and Happiness, After the Digital Explosion* Write a 3 page paper to explore the impact that privacy has on our everyday life. In addition, students will participate in a panel discussion.

Programming - Prime Factorization Students create a program that will find the prime factors of the integer entered. Students use the web to identify an existing code for finding factors. Then students analyze and modify existing programs to determine the best algorithm for prime factoring. Students demonstrate why and how their algorithm is better and more efficient.

Programming Unit Project: Polynomial Factor Program Students create a program to factor polynomials up to degree three. Students start with a greatest common factor detector which they should be able to adapt from their prime factorization program. This

will work for first degree expressions and simplify the harder levels. Students will then add factoring of second degree (quadratic) expressions using a guess and check algorithm. Finally they will use rational root theorem to factor third degree expressions into second degree and use their second degree function to finish. Differences of perfect cubes will make a great platform for reinforcing the need to check for border conditions and special cases. The program that students create must include a gaming theme and be visually attractive.

Skill Practice sets from State Approved Algebra 2 Textbook - In order to create a program that do prime factorization of integers, students must know the algebraic process of factoring integers, therefore students will complete practice problems on prime factorization from the textbook. Also, to write a program that factor polynomials, students must first mastered the skills of factoring polynomials by hand. Hence, students will work on practice problems from the textbook on factoring polynomials.

[Unit 4 - Rational Expressions and Equations](#)

Algebra 2 students are historically weak in this area of mathematics which requires the mastery of factoring multiple polynomial expressions, multiplying/dividing rational expressions, and finding a common denominator to add/subtract rational expressions.

Having mastered the skills of factoring and solving polynomial equations in Unit 3, students use traditional Algebra 2 approaches to simplify rational expressions and solve rational equations prior to modifying their existing Unit 3 factoring program. The modified program, done as one of the key assignments, will simplify rational expressions. Other key assignments will build on their programming skills and concepts of control structures to task students to download and analyze the algorithm of one of the popular fractals programs that is based on iterating rational functions. Students will also explore the applications of rational functions in the creation of fractal images by experimenting with their own input of an iterating rational function. In the next key assignment student will write and run a program that uses rational equations to solve real world problems like rate problems, work problems, distance estimations in projectile motion, percent mixture problem, and Indy car races. This will lead them to explore roots and poles of rational functions, graphs, values for which the denominator is zero, and determining their asymptotes.

Major Topics:

- ★ Simplify rational expressions
- ★ Solve rational equations
- ★ Enhance the programming code from Unit 3 to simplify rational expressions and solve rational equations
- ★ Analysis of Fractal algorithms
- ★ Create programs that use nested conditional and iterative control structures
- ★ Efficient use of math functions built in the programming library for creating programs that solve real world programs
- ★ Apply appropriate algebraic techniques to solve rate problems, work problems,

- percent mixture problems, rational function graphs
- ★ Analysis of rational functions to determine their roots, poles, and locate their asymptotes graphically.

Unit 4 Assignments

Programming Task 1 The Unit 3 Programming Task Key Assignment factored polynomials of degree 2 or 3. Building on the implementation of that key assignment and the additional Algebra 2 knowledge of simplifying rational expressions, students will write and run a program that will: 1). demonstrate knowledge of the use of various operator, operator precedence, data types, variables, and control structures in simplifying rational expressions 2). solve rational equations by multiplying both sides by the LCD to eliminate the denominator. Students will be required to work in cooperative learning groups and their solutions which includes a planned pseudo code, the source code and the output will be presented to the class demonstrating that the program meets the goal of the assignment.

Programming Task 2 - Students will write and run a program that uses rational equations to solve rate problems and distance estimations of the projectile motion or other motions like Indy car race and fluid flow. Students will be required to work in cooperative learning groups and their solutions which includes an a planned pseudo code, the source code and the output will be presented to the class demonstrating that the program meets the goal of the assignment.

Programming Task 3 Students will download a fractal image based on iterating rational functions. They input their own expressions for any rational function of their own in order to help them explore fractals like Mandelbrot and Julia sets. They will analyze the algorithm; identify the type and levels of iteration and its efficiency. Students will be required to work in cooperative learning groups and their written analysis which includes the source code of the fractal and the graphical output will be presented to the class demonstrating understanding of the concepts underlying the creation of fractals and the part rational functions play in particular types of fractals.

Weblog Prompt: "Evaluate and analyze the differences and similarities of the algebraic algorithm of simplifying rational expressions to the algorithm of your programs?" (Approximately 2 pages.) Students will read and respond to 3 peer blog entries.

Skill Practice sets from State Approved Algebra 2 Textbook In this unit, students will review most of the concepts from unit 3 and practice problems from the textbook until mastery in simplifying rational expressions and factoring polynomials. Then, students will work on practice problems from the textbook on polynomials and solving polynomial equations involving rational and complex roots. This will build a solid knowledge base for the students to take on more challenging problems in the textbook and do more practice in multiplying rational expressions, dividing rational expressions, and solving rational

equations. Students will also use the skills and concepts developed in the course thus far, to solve real world problems that use rational equations, explore roots and poles of rational functions, graphs, and determine their asymptotes.

Unit 5 - Exponents and Logarithms

Students receive instructions on operations, modeling growth and decay, and methods of solving exponential and logarithmic functions. Students practice these skills through practice problem sets.

The relationship between exponential and logarithmic functions is an annual stumbling block for students. Through analysis of different search and sort algorithms students will gain clarity of what a logarithmic function represents. As students classify the algorithms as linear or exponential/logarithmic and compare their efficiencies they continue to be exposed to the connections between the computing world and math. Students use programming knowledge and skills from previous units to develop programs that model exponential functions and explore growth patterns, such as loan amortization. Students use array data structures to explore and model exponential growth and decay. Students analyze and create programs that use recursive algorithms to explore and model solutions.

Major Topics:

- ★ Analyze programs to compare different search and sort algorithms
- ★ Use data structures to display loan amortization
- ★ Recognize recursive algorithms and justify their use
- ★ Use exponential functions to explore growth and decay patterns
- ★ Solve exponential equations that model real situations that may include bacterial growth, radioactive decay and population changes

Unit 5 Assignments

Code Analysis Given a sequential search and a binary search, students will analyze the efficiency of each search and classify them as linear or logarithmic. Given a sampling of sort functions, students will analyze the efficiencies of each sort and classify them according to the graph of sample size vs how many swaps. Students will present their code analysis.

Programming 1: Loan Amortization Students create a program that will solve exponential and logarithmic equations demonstrating how interest on a loan is compounded over a long period of time using an appropriate data structure i.e. arrays.

Programming 2: Catch the Fever Write a program to model the spread of a virus. Based on the initial percent of the population who is immune, virulence (percent of people who contract the disease who die), duration of infection (up to 20 days), rate of transmission (how many people contract the disease per day .1 to 10) they will show the percent of people who are sick, immune and total population still alive (assume 100,000 people to

start).

Weblog: Prompt: "Research a programming code that utilizes recursion as a programming technique to solve complex problems. Compare the recursion with problem solving techniques you already know." (Approximately one page.) Students will read and respond to 3 peer blog entries.

Skill Practice sets from State Approved Algebra 2 Textbook - This unit will have practice sets from the textbook in the topics of modeling exponential equations, graphs of exponential equations, and solving exponential equations. Included in the problem sets will be real-world applications of exponential functions such as population growth and loan amortization.

Unit 6 - Systems of Equations

In Unit 6 students build upon the data structures they learned in Unit 5. Instead of data stored in 1-dimensional arrays, students will use matrices (2-dimensional arrays). The matrices will ultimately be used to solve a system of equations entered by the user. To start with, students graph and solve two variable equations with emphasis on elimination method. Once students are comfortable with this method we will introduce them to solving using matrices and row reduced echelon form (i.e. using a graphing calculator). Their foundation in the algebraic and matrix solving methods will come together for the unit project. They will write a program where they have to manipulate the data within the matrix to solve a system of equations. Finally, students will extend their program to handle larger systems (3 or more variables) and compare the methods.

Students use the program they created to help them find the feasible region for a linear programming problem about maximizing profit for a software company.

Major Topics:

- ★ Manipulate more complex data structures (e.g. 2-dimensional arrays)
- ★ Solve systems of linear equations with multiple methods (i.e. algebraically, graphing, matrix on calculator)
- ★ Solve linear programming/optimization problem using technology
- ★ Compare situations in which computers will save time and when it makes more sense to do it by hand
- ★ Change matrices into row reduced echelon form and understand why this yields a solution to the system

Unit 6 Assignments

The assignments for this unit require the student to demonstrate mastery with the Algebra 2 topic of solving systems of equations. Initially, the student will solve systems by the traditional paper and pencil approach. This will lead the student to develop an algorithm that can then be used when writing to program. This meets the UC "c" requirement for the traditional Algebra 2 course along with the CTE goal of programming using 2-D arrays, transitioning from the use of 1-D arrays in the previous unit.

Solve without technology Students complete a series of systems of equations problems using algebraic methods and check their solution by graphing the system. The problems will be selected to encourage use of elimination method which is the basis of solving using matrices.

Solve with technology Students solve systems of equations with graphing calculator using matrices. Students will learn about the format of augmented matrices in order to solve using row reduced echelon form. Upon completion of these problems, students will be ready to start their program.

The Matrix! Students write a program to solve a system of equations in two variables. The students will use the 2-dimensional array to store the equations and manipulate the data based on the elimination method and resulting in a row reduced echelon form matrix. Once their program is proven effective by checking it against their previously solved problems, they extend it to larger systems.

Weblog: Prompt: "Compare and contrast efficiencies of each method of solving the system by comparing which makes sense for one variable, two, ten, etc. and why it is more efficient. This will be the basis of an in class discussion about the uses and limitations of computers in mathematics." (Approximately 2 pages.) Students will read and respond to 3 peer blog entries.

Linear Programming In a linear programming problem, the maximum or minimum value is a corner of the feasible region. The corners are the intersection points of the constraint lines. Students can use the program they created to help them find these points. The theme of the linear programming problem will be maximizing profit for a software company.

Skill Practice sets from State Approved Algebra 2 Textbook - This unit will include practice sets in solving systems of linear equations. Initially the practice sets will include two equations and two unknowns but then will transition to more than two equations with two unknowns. This will tie in well with the programming assignments for the unit. In addition there will be problem sets on Linear Programming, requiring the students to graph a system of inequalities, note the feasible solution region and choose the solution from the intersection points.

Texts:

Title: District approved Algebra 2 textbook.

Edition: Current

Publication Date: Most Recent

Publisher: Any

Author(s): Any

Usage: Primary Text - Read in entirety or near entirety

Title: *Blown to Bits: Your Life, Liberty and Happiness after the Digital Explosion*

Edition: N/A

Publication Date: 2008

Publisher: Online text; see URL

Author(s): Abelson, H., Ledeen, K. , and Lewis, H.

URL Resource: http://www.bitsbook.com/wp-content/uploads/2008/12/B2B_3.pdf

Usage: Primary Text Read in entirety or near entirety

Title: *How to Think Like a Computer Scientist*

Edition: N/A

Publication Date: 2003

Publisher: Online text; see URL

Author(s): Downey, A

URL Resource:

http://openbookproject.net/thinkcs/archive/java/english/dist_v4.0/thinkapjava.pdf

Usage: Primary Text - Read in entirety or near entirety

Supplemental Instructional Materials

Blogspot - <http://www.blogspot.com/m>

This is a free online weblog that will be used in each unit for students to write about their experiences and answers to Weblog prompts (Units 1-6).

Scratch - <http://scratch.mit.edu/>

Scratch is recommended because it allows an introduction to programming in a welldefined, graphical environment that students get up-to-speed quickly. Simple and free programming environment for end of unit programming projects (suggestion: Units 1- 3).

Build Your Own Blocks (SNAP)- <http://byob.berkeley.edu/>

This website has tutorials on how to program more advanced ideas in Scratch. This will be used to teach/challenge students, who completed their projects early, to write more complex programs. (Units 1-3)

Processing - <http://www.processing.org/>

Once students have had some experience in programming in Scratch, Processing offers a graphical programming environment using Java. This programming environment has graphical capabilities and models programming concepts within a visual context because its libraries extend the software to run online sounds, video, and vision. This will help enrich the students understanding. It's a more complex and free programming environment for end of unit programming projects (suggestion: Units 1, 2, 5, 6).

Oracle Java site - <http://www.oracle.com/technetwork/java/index.html>

For those who are interested using the full Java language over a more sheltered environment that Scratch and Processing provide. More complex and free programming environment for end of unit programming projects (suggestion: Units 3, 4).

Featured Projects - <http://scratch.mit.edu/channel/featured>

This website has a library of existing Scratch programs that students can search through to find the perfect program for their Unit 3 research project. (Unit 3)

Applications of Rational Expressions -

http://highered.mcgraw-hill.com/sites/dl/free/0072450282/31462/sect7_8.pdf

Students will use this practice worksheet to solve real world problems involving rational expressions and functions (Unit 4).

Fractal Domains - <http://www.fractaldomains.com/>

This is a program that generates fractal images based on iterating rational functions. Students can generate an unlimited number of fractal images based on iterating rational function while working in cooperative learning groups. Students will focus on the color images generated by the most popular fractal, the Mandelbrot set (Unit 4).

Mandelbrot Set: How it is generated - <http://www.youtube.com/watch?v=8ma6cV6fw24&feature=related>

Introductory materials for fractals that will help students understand how fractal are created by graphically visualizing their unique properties of self-similarities at fractional dimensions (Unit 4).

Scratch Forum - <http://scratch.mit.edu/forums/viewtopic.php?id=61381>

This is a guide on plotting the Mandelbrot Set and explains what is the Mandelbrot Set (Unit 4).

Diseases Activity Simulator -

http://science.education.nih.gov/customers.nsf/HS_Diseases_Activity4_Simulator.htm

This is used as a simulator to model the bacterial growth problems. Students enter initial percent of immunity, virulence, duration of infection, and rate of transmission. Then, this page will run the simulator and provide the students with a chart of information (Unit 5).

Computer Science Unplugged - <http://csunplugged.org/sorting-algorithms>

This is an activity to demonstrate sort algorithm. In this activity students compare different algorithms to sort weights in order from the lightest to the heaviest (Unit 5).

Catch the Fever -

http://www.connectedcalifornia.org/downloads/curriculum/CatchtheFever_CA.pdf

This document has information on dealing with Disease which will help students understand the spreading of a virus for their "Catch the Fever" programming project (Unit 5).

Sorting Algorithms - http://www.youtube.com/watch?v=INHF_5RixTE

This is a video that demonstrate sorting algorithms. By watching this video, students will have a better understanding on how to use sorting algorithms to complete their Code Analysis assignment (Unit 5).

Khan Academy - <http://www.khanacademy.org/>

This website provides practice and review videos to help students, who struggle with Algebra 2 concepts, mastered the concepts taught in class (Units 1-6).