* **Start day of each 4-week unit:**
  + Hand out the 3 project lab/assignment/extra rubrics for the segment
  + 1 project focus per week
* **Weeks 1-3:**
  + Day 1:
    - First half of class:
      * Explain code required to complete the lab
      * Teacher walks through the code (on projector/screen) that will be completed during lab time (will start with a partially completed code – e.g. fill in methods. In some cases will start with a blank file.)
    - Second half of class:
      * Lab time: start on your lab project (alone & direct teacher help)
  + Day 2:
    - First half of class:
      * Continue explaining code required to complete the lab
    - Second half of class:
      * Lab: finish/review on your lab project (pairs & direct teacher help)
    - Lab project is due at end of day (turn in what you have or come back at end of school day to finish with the teacher)
  + Day 3:
    - Assignment day (assignments are not due until the beginning of the coming week, so can be completed during class on this day or over the weekend).
    - Review/study-hall/etc.
    - Can be used to talk about general computing topics like UNIX, WWW, security, OOP, etc.
    - Suggested reading announced: this class period can be used for reading sections of the textbook(s)
* **Week 4:**
  + Day 1:
    - Review day (hand out the corresponding FRQ and MCQ problems for students to work through with direct teacher help)
  + Day 2:
    - Test day (changed version of FRQ and MCQ problems)
  + Day 3:
    - Extra credit day (extra credit can be turned in at any time before end of semester)
    - First half of class:
      * Review the extra credit requirements from the past 4-week segment
    - Second half of class:
      * Study hall or students can get help on extra-credit work

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Week | Physics Track | Data Manipulation Track | I/O & Network Track | Tutorial Track (PPT or Markdown) |
| 1 | 1-3 | Lab: **Software (JDK & Atom),** **introduction to variables, functions, and print-output**: calculate a ball trajectory distance: solve equations by hand, then write program using only variables.  Assignment: calculate the height reached by the ball: **solve by hand**, then add height to program from lab.  Extra: height/distance for 4 independent balls | Lab: **introduction to computer architecture**: simplistic “compilation” for trivial processor: (1+2)\*(3+2), then loop to increment variable to a threshold.  Assignment: draw a logic circuit for a 2-bit adder (ALU of the trivial processor) and write program instructions to execute (2+2)/(1+1).  Extra: draw logic circuit for a 4-bit adder. | Lab: **introduction to UNIX interface**: use command line to navigate to .java, compile using javac, and run using java.  Assignment: create a simplistic .java program that prints something, and then record <1 min (narrated) screen video of navigating to the directory, compiling, and running the .java file. | Research:   * Primitive variables * Operators and Boolean logic * If-then statements * Loops * Strings & String literals |
| 2 | 4-6 | Lab: **introduction to loops and arrays**: write Physics.java with time(), distance(), and graph(), and test using TestPhysics.java.  Assignment: change each while-loop to a for-loop; add a Physics.graph() method that prints array using ‘\*’ characters; change TestPhysics to use the .graph() method.  Extra: ###### | Lab: **introduction to classes, toString(), and “this”**: write GraphModel and Line classes. write TestGraphModel class that creates two GraphModel objects, plots y=x^2 from -10 to 10 on one model and loads 2 stand-alone Lines on the other, and print both models using toString().  Assignment: 1. write comments (in your own words) for each method. 2. add getLines(), minX(), minY(), maxX(), maxY(), and endPlot() methods. 3. Change toString() of GraphModel to only print Line objects, and change output to CSV compatible. 3. Copy/past output into CSV file and graph using Excel.  Extra: In TestGraphModel, write code/method that creates new lines between pairs of parabola Points (horizontal lines). | Lab: **introduction to UNIX I/O, CLI arguments, and Exceptions**: 1. Receive string arguments from the command line, 2. Check for zero-length, 3. Print each argument or message that no arguments exist.  3. Change program to do the following: 1. If an arg is preceded by the arg “int”, then 2. convert to an int, 3. Use a try-block to multiply the int by the previous int and print, or 4. Print a message that the int couldn’t be converted.  Extra: Re-write the Physics and Data Test### classes to receive variables from the command line arguments output to CSV file. Graph using Excel. | Research:   * Arrays * Multi-dimensional arrays * Classes * “this” * String.equals() * String.length() * Try-catch blocks * Java Exceptions |
|  | 7 | Test 1: one of the 2016 FRQ questions (changed slightly) + three 2014 MCQ questions (changed slightly) |  |  |  |
| 3 | 8-10 | Lab: Write the Physics.trajectory() method (based on code from ThrowBall.java in module 1) and test in TestPhysics.java using the .graph() method.  Assignment: re-implement ThrowBall.java to use Physics.java to print trajectories of 4 different balls using A,B,C,D. (0.1s interval). Optional: throw a ball in front of a grid on the wall, record a movie, and loop through possible velocities to match distance (determine angle and initial height from movie).  Extra: Loop through angles for each of your 4 balls (differing heights and velocities), and find optimum angles. | Lab: **Introduction to interfaces and for-each**: implement GraphView interface as GraphViewText class. write ScreenTransformation class and a GraphController class that instantiates GraphModel and GraphViewText objects, calls plotLine() with 4 Points (creates 3 Lines), and shifts the 1st intersection of the lines (creating a zig-zag).  Assignment: 1. write comments (in your own words) for each method. 2. draw axes in the refresh() method 3. Convert each for -loop into a for-loop (where possible) **learn different uses for for vs. for-each**. 4. Re-write the drawLine() method to plot x(y) instead of y(x) for slopes greater than 1, 5. Plot y = -0.02x^2 + 2x + 200, at resolution (50,50).  Extra: draw lines from the origin to each point in the parabola. | Lab: write a simplistic program to use Scanner and System.in to obtain a string input from the user (print result).  Assignment: add another input for an int to the program  Extra: parse one line containing both string and int data. | Research:   * Model-View-Controller model * Interfaces * Abstract classes |
|  | 11 | Test 2: one of the 2016 FRQ questions (changed slightly) + three 2014 MCQ questions (changed slightly) |  |  |  |
| 4 | 12-14 | Lab: **introduction to polymorphism and ArrayList<>**: extend Point as Vector, create a FreeBody class and extend as Rocket class. Test Rocket with TestRocket class. Display height, velocity, distance curves by calling toString() on the location Vectors.  Assignment: 1. copy Point, Line, GraphView, GraphViewText, GraphController from the Data track, 2. Instantiate model/view to display locationY(locationX) and model/view to display locationY(time).  Extra: ######## | Lab: **introduction to Java graphics**: Write the GraphViewGUI class to implement GraphView using a GUI OS window. Run GraphController and display parabola in GUI window.  Assignment: re-graph Physics track module 4 using GraphViewGUI.  Extra: graph a 3rd order function and a sin wave. | Lab: **introduction to File objects**: write a program to read from a text file using Scanner and read into a 2D array. Write the text file to a CSV file.  Assignment: transpose and print 2D array and write to a CSV file.  Extra: output the transposed array as an HTML table, and pipe to an .html file. |  |
|  | 15 | Test 3: **two** of the 2016 FRQ questions (changed slightly) + three 2014 MCQ questions (changed slightly) |  |  |  |
| 5 | 16-18 | Lab: change the Ball class to **iteratively** calculate x,y of a ball thrown from a non-zero height. **Each new vector** is calculated by the computer based on the vector from the previous iteration. (not solved by hand).  Assignment: iteratively track 10 balls.  Extra: incorporate a reverse drag vector. | Lab: write a program to read text and create a percentage for each word: (word\_sum / total\_words). Introduce bubble sorting. Print out top 100 word & % for multiple authors & texts. Pipe to CSV file and plot using Excel.  Assignment: import two text files, use only top 100 words that appear in both, and pipe to CSV file.  Extra: import two files from two authors, use only common words, pipe to CSV file. | Lab: program to read all the files in a directory into an ArrayList and create a file containing lines with the name and length of each file.  Assignment: change program to combine all files and save to one file in a different directory.  Extra: output to an HTML document with scrollable DIV elements displaying each file |  |
|  | 19 | End of first semester:  Test 4: **two** of the **2017** FRQ questions (changed slightly) + three 2014 MCQ questions (changed slightly) |  |  |  |
| 6 | 20-22 | Lab: built-in classes and interfaces: ArrayList<Ball> and interface & implementation for a text-graphing class.  Assignment: Use an ArrayList<> to store X number of Balls, and graph using the graphing class.  Extra: add axis numbers to the graphing implementation | Lab: HashMap of Distribution objects (containing mean, min, max, std\_dev methods) for each top 100 (in common) word. Read from a directory containing at least 10 texts by an author. Print CSV file with word,avg,min,max,std\_dev.  Assignment: read from 3 directories (3 authors) and score 1 for each word that is less than 1 std\_dev from the other author. Print the score / total\_common\_words as a percent when comparing each pair of authors.  Extra: compare books of the bible in original Hebrew/Greek (provided) and evaluate whether there is strong evidence for books with common authorship. | Lab: Open a ServerSocket and respond to a GET request with a simple HTML string. Test functionality using a web browser.  Assignment: parse the GET string for a filename only (ignore directory paths), and use that to open a file and print that to the client.  Extra: allow GET string to include full directory filepaths (removing any “..”) for security. |  |
|  | 23 | Test 5: **two** of the **2017** FRQ questions (changed slightly) + three 2014 MCQ questions (changed slightly) |  |  |  |
| 7 | 24-26 | Lab: extend the Ball class to create a Rocket class with a thrust curve and duration. Add a getThrust method. Iteratively calculate trajectory of the Rocket during thrust and in glide.  Assignment: launch and track 10 rockets.  Extra: add a reverse drag coefficient, and have the rocket deploy a parachute after apogee (very high drag coefficient) |  | Lab: Open a client socket to connect to the server and send a GET string. Receive the string from the server and save to a file. Test functionality by connecting to web site and downloading the HTML document.  Assignment: parse the returned HTML and print the document title (along with saving the file).  Extra: parse returned HTML for table data, and save table data to a CSV file. |  |
|  | 27 | Test 5: **two** of the **2018** FRQ questions (changed slightly) + three 2014 MCQ questions (changed slightly) |  |  |  |
| 8 | 28-30 |  |  | Extra: compare unicode text of several Bible books in (original Hebrew/Greek -- to be provided) and evaluate whether there is strong evidence for books with common authorship. |  |
|  | 31 | exam is **two** of the **2018** FRQ questions (changed slightly) + three 2014 MCQ questions (changed slightly) |  |  |  |
| 9 | 32-33 |  |  |  |  |
|  | 34 | End of **AP Year**:  exam is **ALL** of the **2019** FRQ questions (changed slightly) |  |  |  |