

# Computational Methods in the High School Physics Classroom

A Major Qualifying Project

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

DEGREE OF BACHELOR OF SCIENCE

By:

Xavier Morales

March 2024

*This report represents the work of one or more WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on the web without editorial or peer review.*

## **ABSTRACT**

Programming is an increasingly important skill in both physics research and physics-related industries. However, computational methods are often separated from physics curriculums at the university level, and are scarcely learned at all at the high school level. My project was to develop programming lesson plans that could be incorporated into high school physics curricula in accordance with the Massachusetts DESE standards. Through extensive research into both teaching philosophies and specific pedagogical tools, as well as extensive field testing in collaboration with experienced teachers, these lessons and their accompanying materials provide a solution for integrating physics and computer science at the high school level.

# Table of Contents

<b>ABSTRACT</b> .....	<b>2</b>
<b>ACKNOWLEDGMENTS</b> .....	<b>4</b>
<b>REAL WORLD EXPECTATIONS AND THE HIGH SCHOOL PHYSICS CURRICULUM REALITY</b> .....	<b>5</b>
<b>BRIDGING THE GAP BETWEEN PHYSICS AND COMPUTER SCIENCE</b> .....	<b>7</b>
WHAT IS LEARNING?.....	7
BEST PRACTICES FOR EFFECTIVELY ASSESSING LEARNING .....	8
<b>METHODOLOGY</b> .....	<b>10</b>
UNDERSTANDING BY DESIGN .....	11
STATE STANDARDS AND LEARNING GOALS .....	12
LESSON DESIGN PROCESS .....	13
<b>FORMAL LESSON DESIGNS</b> .....	<b>16</b>
LESSON PLAN 1 - INTRO TO SPREADSHEETS.....	16
LESSON PLAN 2 - KINEMATICS AND 2-D MOTION IN GOOGLE SHEETS.....	19
LESSON PLAN 3 - SIMPLE HARMONIC MOTION IN PYTHON.....	21
LESSON PLAN 4 - ELECTROSTATIC AND GRAVITATIONAL FORCES.....	23
<b>DISCUSSION</b> .....	<b>26</b>
FIELD TESTING.....	26
COLLABORATIVE EFFORTS.....	29
<b>FUTURE STUDIES</b> .....	<b>30</b>
<b>APPENDIX A – COMPLETE LESSON PLANS</b> .....	<b>33</b>
LESSON PLAN 1 - INTRO TO SPREADSHEETS.....	33
LESSON PLAN 2 - KINEMATICS AND 2-D MOTION IN GOOGLE SHEETS.....	37
LESSON PLAN 3 - SIMPLE HARMONIC MOTION IN PYTHON.....	40
LESSON PLAN 4 - ELECTROSTATIC AND GRAVITATIONAL FORCES.....	43
<b>APPENDIX B – STUDENT PACKETS</b> .....	<b>47</b>
PRE-CLASS REVIEW: INTRO TO SPREADSHEETS.....	47
INTRO TO SPREADSHEETS: PROJECTILE MOTION ACTIVITY.....	49
TAKE-HOME ASSIGNMENT .....	50
<b>APPENDIX C – IDEAL CODE</b> .....	<b>72</b>
LESSON 3 .....	72
LESSON 4 .....	73
<b>APPENDIX D - RUBRICS</b> .....	<b>74</b>
LESSON 1 .....	74
LESSON 2 .....	75
LESSON 3 .....	76
LESSON 4 .....	77
<b>REFERENCES</b> .....	<b>78</b>

## **Acknowledgments**

I'd like to thank my advisor, Professor Thomas Noviello, for his continued support and constant encouragement throughout the completion of this project and my undergraduate career. He played a key role in the inception of this project and provided meaningful guidance throughout every step thereafter. The final product as it stands would not have been possible without him.

I'd also like to thank all the high school teachers in the Teacher Advisory Group, especially Rob Gendron, Maria de Vuono-Homberg, Avery Travis, as well as Ben Petkie and Thomas Young, for their kind words and immense help with testing my lessons. They provided an important perspective that uniquely highlighted the potential for this project.

## **Real World Expectations and the High School Physics Curriculum Reality**

The rise of computer technology over the last century has revolutionized the scientific world. Programming in particular has contributed to projects such as the moon landing, the development of nuclear bombs that ended World War II, and various other important scientific developments of the 20th century. As described in an article by the American Association of Physics Teachers (AAPT, 2016), programming today plays a role in “almost all areas of physics research.” At the university level, a basic understanding of programming is expected of all aspiring physicists, and after university, it is possibly the single most important skill to have in order to obtain any physics-related job. According to the American Institute of Physics (McNeil & Heron, 2017), almost 60% of physics graduates work in STEM or Computer Science, and of these over 80% cite programming as a regularly used skill. However, there is a severe lack of computational instruction in current physics curriculums.

The two aspects of computer science that apply to physics are general programming and modeling. In both cases, there are problems with how current university programs incorporate them. For many physics students, basic programming skills are prerequisites to build computational physics skills. The AAPT asserts that while some have extensive experience coming into college, “others have never written a single line of code.” For these students, there exist introductory computer science courses. Due to various challenges such as time constraints, inadequate instructor backgrounds, and scheduling issues, implementing basic programming directly into physics courses has not been popular (AAPT, 2016). At the same time, modeling instruction is limited in computer science courses. While physics courses often develop modeling skills, the necessary background knowledge is overlooked. This system of expected knowledge

with limited instruction leads graduates to have a general dissatisfaction with their programming knowledge. According to the AIP, physics graduates in a 2017 study were “virtually unanimous in their desire for more programming skills,” (McNeil & Heron 2017). A potential solution to this would be to increase computational instruction at the high school level.

A significant portion of university students studying physics arrive at college with minimal programming experience. According to a 2012 report from Georgia Tech, about “half” of their introductory mechanics students had zero programming experience. The other half had received some level of prior instruction either in high school or in previous semesters at the university (Caballero et al. 2012). In a blog written that same year, University of Michigan Professor Mark Guzdial estimated that less than 10% of high schools offered any computer science courses (Guzdial 2012). This indicates that the percentage of students entering college in 2012 with any level of programming experience was minimal. 11 years later, the Computer Science Teachers Association (CSTA) reported that up to 58% of high schools in America offered computer science classes. However, they went on to report that less than 6% of high school students nationwide were enrolled in these courses (Hendrickson et al., 2021). While improvements are evident over the last decade, there still exists a major gap in the adoption of computer science courses at the high school level.

My goal is to design programming lesson plans for the high school physics classroom that effectively expose students to the world of computer science while adhering to the physics frameworks set by the Massachusetts Department of Elementary and Secondary Education (DESE). These lessons are meant to be implemented directly into existing physics curricula, and were designed to work under specific physics standards and objectives. Each lesson will build on the last, and will together teach students a level of programming sufficient for them to find more

immediate success at the college level and beyond. In order to measure the success of the lessons in achieving this goal, a method of assessment must be established that will both track and encourage student learning.

## **Bridging the Gap Between Physics and Computer Science**

### *What is Learning?*

Backwards-looking assessments, such as the SAT or ACT, test whether students have learned a particular topic by asking them to recite information (Fink, 2005). In these traditional examinations, knowledge is the key target being assessed. To analyze the effectiveness of this style of assessment, let us first assume that knowledge and learning are equivalent. If this were the case, the most prepared person in the world would be the individual who absorbed, and remembered, the most information. The main issue with this statement is that unless they actually understand any of it, memorizing a lot of raw information won't get them very far. Comprehension, then, is an integral piece of the equation. In my personal opinion, learning is a combination of both comprehension *and* knowledge.

To address effective assessment methods, several theories around the nature of learning have been developed. Constructivism proposes that learning is a process through which observations of the real-world lead to constant modification of a student's cognitive landscape. Based on this theory, knowledge and comprehension are established simultaneously. Social constructivists suggest that social interactions in early stages of life create a baseline of knowledge, namely cultural beliefs and ideas, that become necessary for teachers to account for. The Multiple Intelligence theory takes this further by arguing that students may have varying cognitive

structures on a physiological level that affect how they process instruction (Nguyen, 2013). As a result, the “best” way to teach is individual to each student. While no single theory has been determined as the most effective, they all point to the importance of comprehension as well as knowledge.

As stated previously, traditional exams test students on their knowledge of a topic. Based on the above definition of learning, they are therefore incomplete assessments. This then brings into question the true goal of teaching: creating a perception of success by achieving administrative standards, or encouraging the growth of students for the benefit of their future endeavors. If the answer were the former, this method of assessment is perfect. If students are able to successfully regurgitate information, instructors have evidence that they were able to meet their main objectives. On top of being clear indications of how well teachers adhered to the necessary curriculum, these types of assessments are generally easy to design. Nevertheless, I believe that knowledge and learning are in fact *not* equivalent, meaning that backwards-looking assessments do not accurately measure whether a student has truly learned anything. In reality, they promote having short-term memory of information rather than an actual understanding of the ideas being presented (Simonson et al., 2000). As a result, an additional method of assessment must be considered.

### *Best Practices for Effectively Assessing Learning*

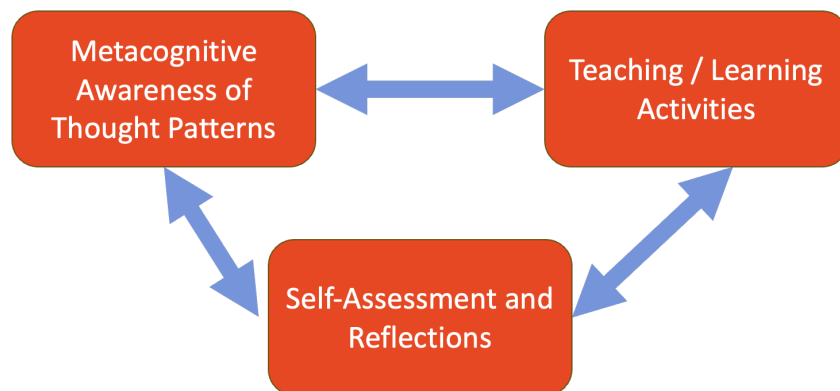
Forward-looking assessments test how well students are equipped to utilize classroom material by anticipating practical applications that may arise in the future (Fink, 2005). Since learning is a unique endeavor for every student, although potentially motivated by different factors



according to the above learning theories, forward-looking assessments are developed with the students in mind. Rather than creating multiple choice questions that ask students to recall specific points of a lecture, these assessments aim to promote learning by encouraging students to apply their knowledge to broader, real-world scenarios. Open-ended questions, essays, and projects are all examples of forward-looking assessments (Fink, 2005). Besides their general emphasis on student learning, these types of assessments are also advantageous due to their introduction of other forward-looking instruction methods such as surveys and self-reflection opportunities. These methods are particularly beneficial to teachers because they provide informative and continuous feedback.

A major downside to traditional assessments is their lack of nuance. While they are generally sufficient in tracking whether a teacher covered particular topics, poor student performance does not provide any indication of a potential root cause (Law & Eckes, 1995). Introducing surveys and reflections as part of a forward-looking strategy, however, indicates gaps in instruction much more effectively. For example, reflections by students can provide insight on their level of comprehension and confidence after attempting to apply knowledge in a practical setting. In the same way, surveys containing open-ended questions provide students the opportunity to give honest feedback on their progress without the pressure of getting a bad grade. By gathering and reviewing data from these strategies, teachers can adjust their lessons and / or instruction strategies to mirror student needs. In this way, teachers can more effectively determine how to best support their students (Winking, 1997). Implementing these methods essentially creates a feedback loop for teachers to improve their instruction, which will in turn aggregate greater student learning. At the same time, offering self-assessment opportunities can be beneficial to students by encouraging meta-cognition.

The development of learning theories was largely motivated by a desire to understand students' internal cognitive landscapes so as to optimize learning, specifically from the teacher's perspective (Nguyen, 2013). This internal landscape is where learning often occurs on a subconscious level, but it can also be harnessed and used as a tool by the students themselves. According to a research paper by Kimberly Tanner, "Students will not really learn new information if they do not go through a metacognitive realization that requires them to examine how they thought about a topic before and how they are thinking differently about the topic now" (Tanner, 2012). In other words, recognition and awareness of how one learns is just as essential as effective instruction for fostering learning. This awareness, commonly referred to as meta-cognition, can be cultivated through student reflection and self-assessment. Just as with the principles behind forward-looking assessments, the emphasis on tangible learning reflected by self-regulation methods was an integral aspect of the lessons, effectively acting as a pedagogical backbone in their design.



*Figure 1: Learning-centered Education Model*

## **METHODOLOGY**

## *Understanding by Design*

Backwards design is an educational development framework that promotes a reverse-engineering strategy to course creation. This framework was first written about by Grant Wiggins and Jay McTighe in their book named *Understanding by Design* and was later espoused by the Vanderbilt University Center of Teaching. The central dogma of backwards design is that creating lessons should be a top-down process, beginning with the establishment of overarching goals and finishing with the lessons themselves. By starting with the highest level of abstraction before moving into the practical details, the resulting lessons are specifically learning-focused rather than teaching-focused. As described by Professor Ryan Bowen in a Vanderbilt article, the main benefit of adopting a backwards design strategy is that it “encourages intentionality during the design process” (Bowen, 2017). To the students’ benefit, this intentionality fosters learning by emphasizing not only *what* is being taught, but *why* it is being taught as well.

The three main stages of backwards design, as described by Professor Bowen, are identifying the desired results, determining acceptable evidence, and planning learning experiences and instruction (Bowen, 2017). The first stage involves separating info into three groups - knowledge worth being familiar with, knowledge that is important to do what has to be done, and knowledge that is necessary for enduring understanding. The second stage of backwards design is choosing methods of assessment that will best align with the knowledge most necessary for achieving the overarching objectives. The final stage of backwards design is the lesson plan creation. In creating my programming lessons, the backwards design process began with the establishment of a philosophical perspective centered around the definition of learning. The next

step was the development of learning objectives based on specifically chosen Massachusetts High School Physics Standards.

### *State Standards and Learning Goals*

The Next Generation Science Standards (NGSS) are an educational framework that has become widely adopted across the United States in recent years. Created as a way to reorganize and refocus the national education system, the NGSS were built around the idea of three-dimensional learning. Specifically, the three dimensions are practices, crosscutting concepts, and disciplinary core ideas. Practices are defined as the behaviors of scientists in their investigation of theory/model-based inquiries or of engineers in their design process for solving real-world problems. Crosscutting concepts are learnable ideas that can be subsequently applicable across areas of science. Disciplinary core ideas consist of the most essential science teachings such that students will be able to obtain a “complete” scientific knowledge base (<https://www.nextgenscience.org/>). While Massachusetts did not officially adopt these standards, they used them as a major influence when revising their own standards. As a result, Massachusetts Education Standards share many of the same principles as the NGSS.

In order to be implemented into existing physics classrooms, my lessons were built around both NGSS and Massachusetts High School Physics Standards. By providing computer science instruction through the lens of physics, the lessons strongly lean into the idea of crosscutting concepts. Further, creating a base of programming knowledge will be useful, and often essential, in almost any career path. As a result, the skills taught in my lessons also fall under the categories of practices and core ideas. From a more practical perspective, the lessons each follow a particular

Massachusetts Standard so that they can be easily added into existing physics curriculums. Instead of being extra work for both teachers and students, they will compliment and even replace certain lessons in accordance with state guidelines. Once standards were chosen, the next step was to develop learning objectives.

Creating useful learning objectives involved matching sets of programming concepts with certain physics topics. In each lesson, a physics objective was created based on a corresponding standard. Each physics objective was based on a particular topic that has historically yielded high school lab opportunities. This way, coding activities could replace potential lab opportunities instead of taking the place of a lecture or discussion-based lesson. From here, computer science objectives were constructed to specify what programming concepts would be the foundation of each lesson. Once the objectives were set, I was able to begin creating lessons that were both anchored in my philosophy and driven by my objectives.

### *Lesson Design Process*

Each lesson plan follows the same general format: pre-class preparatory work, a hook activity, a brief teacher presentation, a group activity, self-assessment using a rubric, and a post-activity reflection. Since these are just individual lessons, pre-class work was necessary to combat the time constraints associated with only having a few class periods to learn basic programming. The in-class structure is relatively standard, although it is structured more similarly to a lab than a normal class. While developing and applying this format in each lesson, considerable research was done into education best practices. Specifically, these lessons incorporate differentiated instruction, student-centered learning strategies, and student affect assessments.

The idea behind differentiated instruction is that students, whether they have an IEP or not, require varied learning accommodations. As these lessons were made to be disseminated into heterogeneous classrooms across Massachusetts, it was important to address some of the most common accommodations in an effort to make them more easily adaptable. According to an article on the Behaviorist.com, the most common IEP accommodation is alternative instruction (Cummings 2023). The second most common is digital resources. The latter was practically built into the lessons as they all involve online work. The former, therefore, became a focus in the lessons and came in the form of demonstrations, presentations, and group work. Ideally, including various instruction methods will allow for smoother lesson integration into existing classrooms. In reality, making sure each instruction method is optimally designed is just as important as their inclusion. Besides designing slideshows for the teacher presentations and developing demonstration instructions, their execution is largely up to the instructor's individual style. Group work, however, is a bit more complex.

Historically, group work has been used extensively as a way to emphasize student-based learning. By pairing students into small groups of three or four, students are encouraged to lean on and learn from their peers, rather than from an authority figure. Often, however, there exists significant variability in what relevant past experiences students have related to a new concept, which in turn can further polarize a classroom due to induced student passivity. In order to combat this, there are several factors that go into designing successful group work sessions.

In a 2014 study done at the Linköping University of Sweden, researchers discovered three main abstract concepts that marked successful group work (Eva 2014). While these abstractions were relevant to the general field of education research, the study revealed a more intriguing perspective that could be referenced for practical pedagogy: the qualities students themselves

thought to be most indicative of “good” group work. According to the students interviewed in the study, these factors were as follows: “(a) the group work is carried out in supportive context, (b) cooperation occurs, (c) the group work is well-structured, (d) students come prepared and act as working members during the meetings, and (e) group members show respect for each other,” (Chiriac). In order to account for these factors, specific measures were directly incorporated into the lessons. For example, pre-class readings / thinking questions were intentionally placed so that students would have comparable levels of pertinent knowledge. This would ideally increase the preparation, structure, and cooperation of the subsequent group work.

Another way in which student-based learning is emphasized is through the use of self-regulation. By providing detailed rubrics to the students, they would be able to determine progress and completion as they work through the activities. According to a 2017 article from Indiana University–Purdue University Indianapolis (IUPUI), rubrics have many benefits (Gregory & Rehak 2017). For the students, rubrics can enhance the quality of self and peer-assessment as well as encourage them to think critically by linking assignments with learning objectives. For the instructors, rubrics can increase transparency, efficiency, and consistency of grading. In order to obtain these benefits, this article goes on to cite several best practices that were incorporated into the lessons. These include integrating relevant learning objectives as well as using the same language across rubric descriptions and assignment prompts.

The last major component of each lesson is the post-lesson self-reflection. Including these self-reflection opportunities was critical, as they served three distinct purposes. The first was to encourage meta-cognition. Referred to by Tanner as “retrospective self-assessments,” reflections can improve the learning process in students by guiding them to acknowledgement of their own improvements (Tanner, 2012). As a result, the reflection prompts were particularly phrased so as

to direct students towards any thought pattern changes they may have experienced. The second purpose of the self-reflections was to act as a progress checkpoint for the instructor to observe. The third purpose was to provide qualitative data for myself when implementing the lessons in classrooms. These student reflections, as well as interviews with several experienced educators, were essential to the final steps of the design process: field testing the lessons, analyzing the results, and identifying potential improvements.

## **FORMAL LESSON DESIGNS**

### *Lesson Plan 1 - Intro to Spreadsheets*

#### Massachusetts HS Physics Standard:

HS-PS2-10 (MA): Use free-body force diagrams, algebraic expressions, and Newton's laws of motion to predict changes to velocity and acceleration for an object moving in one dimension in various situations.

#### Learning Objectives:

1. Physics
  - a. Students will be able to communicate an understanding of velocity, speed, acceleration, mass, inertia, weight, displacement (change in position), and distance.
2. CS
  - a. Students will be able to identify the main aspects of the google sheets window.
  - b. Students will be able to use the basic cell-based functions available in google sheets, including insert, delete, drag, highlight, fill, etc.
  - c. Students will be able to write arithmetic formulas in google sheets that utilize built-in commands in order to do basic math operations.



- d. Students will be able to identify what graphs are best suited for different types of data sets, as well as understand how to create these different graphs using given tables.
- e. Students will understand and know when to use conditional formatting (filter) as well as conditional formulas (IF statements).

Summary of Tasks / Actions:

- Hook activity: Split class into small groups and give them the following questions to discuss
  - Have you used google sheets in the past? If so, what have you used it for?
  - How could google sheets be helpful to physics researchers?
- The first section of this lesson will be a guided lecture
  - You will go over the following aspects of the google sheets window to begin:
    1. Creating a file
    2. Clipboard (cut/copy/paste)
    3. Font tools (color, size, style, etc.)
    4. Alignment Tools (wrap text, merge, etc.)
    5. Insert tab (tables, pictures, charts, graphs)
    6. Arithmetic operators (reference cells/groups of cells)
    7. Drag-down feature
  - Students can follow along with all of these using personal devices
  - Students can also take notes on these functions, as any of them could prove useful in future uses of google sheets software
- The second section of this lesson will be an activity to reinforce the above skills - Basic Kinematics Problem
  - Give students a 2-D motion scenario (such as a ball getting dropped off a cliff)
  - Given values
    - Acceleration due to gravity
    - Time it takes for ball to hit the ground

- Ball is dropped from rest
- Part 1 - Put values into google sheets and write a formula where you reference the correct cells to calculate the distance the ball falls off the cliff
- Part 2 - Create a new sheet and use your formula again but create a table of time values from 0 to given time in intervals of 0.1 seconds and how far the ball has fallen at each time. Use the drag-down feature to complete this. Then create a graph based on this table that represents displacement vs time with all of your data points shown on the graph. Export and save the graph as a separate file and submit both files to complete assignment.

Take-home tasks:

The take-home work after this lesson will be preparing for the google sheets lab activity that will augment the Kinematics and 2-D motion portion of the physics class. This take-home assignment will essentially be a pre-lab review of both physics concepts and the essential google sheets skills that will be applied in the lab. While everything covered in this lesson is essential to having a foundational knowledge of google sheets, there are a few specific features that are specifically beneficial when working with physics, which include creating graphs, editing graphs, arithmetic operations, and drag-to-fill. These will be used extensively in the lab activity, and a solid understanding of these particular topics will be very important for collaborative, student-based learning to be successful in the following lesson.

*Lesson Plan 2 - Kinematics and 2-D motion in google sheets*

Learning Objectives:

1. Physics

- a. Analyze and compare the motion of objects using position/distance vs. time, velocity/speed vs. time, and acceleration vs. time graphs.

2. CS

- a. Students will be able to manipulate tables using arithmetic functions and drag-to-fill.
- b. Students will be able to create graphs from tables with correct axes labels and titles.
- c. Students will be able to add relevant trend lines based on the scenarios provided by different data tables.
- d. Students will be able to customize graphs in google sheets using advanced formatting features.
- e. Students will be able to export graphs from google sheets to be used in separate documents.

Summary of Tasks / Activities:

- Hook Activity: Projectile Motion
  - Get a volunteer and throw a ball in the air across a distance
  - Then throw a ball straight up and down
    - Ask students what the difference is
    - Based on responses, mention the idea of horizontal velocity having no acceleration, and vertical motion following the same path in both examples
  - Demonstrate the parallel with one final demonstration, but rather than thrown the ball at an angle, like in traditional projectile motion problems, slide across the floor in a chair and toss the ball straight up and down
    - The ball will follow a projectile path, even though you seem to just throw it perpendicular to the ground

- Give students packet to follow
  - Provide link to position time chart in both x and y directions
  - Students are tasked with taking this chart and creating velocity-time and acceleration-time charts for both directions in separate sheets
  - Students will then graph each chart and customize them however they would like
    - Necessary graph details
      - Labels
      - Units
      - Title
      - Correct axes
      - Correct graphs
- Teacher will mostly just monitor the progress of different groups
- Groups will be split up based on previous knowledge on google sheets
  - Groups will be 2-3 people at most, and while they will be working together to get through the parts of this activity, each student will be following along on their personal devices
  - Each student will submit their own work

Take-home tasks:

Since Google Sheets is available on any personal device, students would be expected to complete any part of this activity at home that they did not get to in class. As this lesson is collaboration-based, this would encourage students to communicate outside of class and establish a group dynamic that emulates what it will be like in college.

## *Lesson Plan 3 - Simple Harmonic Motion in Python*

### Massachusetts High School Physics Standard:

HS-PS2-10(MA). Use free-body force diagrams, algebraic expressions, and Newton's laws of motion to predict changes to velocity and acceleration for an object moving in one dimension in various situations.

### Learning Objectives:

1. Physics
  - a. Students will understand the principles of Simple Harmonic Motion and apply them to graph aspects of spring-based motion.
2. CS
  - a. Students will be able to create a .py file, save it, compile it, and run it.
  - b. Students will be able to create graphs in Python using built-in plotting functions.
  - c. Students will be able to properly label graphs in Python.
  - d. Students will be able to export graphs for use in external files.

### Summary of Tasks / Activities:

- Before class: packet to follow as an introduction to Python IDE
  - Basic window features only, as well as how compiling and running a program works.
  - Functionality of Python as a programming language will be described mainly as a parallel to what was taught in google sheets
    - Variable declaration and data types
    - Arithmetic functions
    - Function calls
    - Comments
  - "Hello, World!" program

- Hook Activity: Model a Spring Systems
  - Teacher will begin this lesson showing students a system where a block is oscillating on a frictionless surface while attached to a spring with a known spring constant
  - Students will be tasked to model this system and the different types of energy associated with it
    - **\*\*this computational lesson should be following a broader introduction to SHM, including an overview of the position, velocity, and acceleration of an oscillating system\*\***
  - Models will be a springboard to create a set of energy equations to describe this system
- With a base of knowledge in programming concepts from google sheets, and the basics of Python from the pre-class review, students will be tasked with creating a basic program that simulates an SHM system and create graphs for several components
  - As in the previous google sheets activity, students will be split into groups to complete this project
  - Students will be given values for mass, spring constant, and amplitude
  - With these values they should write a program that uses the SHM equations established at the beginning of class to create graphs for kinetic energy, potential energy, and total energy

*Lesson Plan 4 - Electrostatic and Gravitational Forces*

Massachusetts High School Physics Standard:

HS-PS2-4. Use mathematical representations of Newton's law of gravitation and Coulomb's law to both qualitatively and quantitatively describe and predict the effects of gravitational and electrostatic forces between objects.

Learning Objectives:

1. Physics

- a. Students will be able to analyze and compare systems of two objects with different charges and distances apart to compare the magnitude of the electrostatic forces.
- b. Students will be able to analyze and compare systems of two objects with different masses and distances apart to compare the magnitude of the gravitational forces.
- c. Students will be able to compare and contrast the graphs of gravitational force and electrostatic force to better understand the forces of our universe.

2. CS

- a. Students will be able to develop multiple methods to provide functional procedures in their code.
- b. Students will understand what object-oriented programming is, and be able to explain how the development of methods is a part of this particular strategy.
- c. Students will be able to program loops to calculate data points for their plots.
- d. Students will compare and contrast the graphs of gravitational force and electrostatic force

## Summary of Tasks / Activities:

- Hook Activity:
- Similarly to in the second google sheets lesson, the teacher will act more as a supervisor over student groups
  - This lesson should be implemented at the start of the Electricity and Magnetism unit
  - Students at this point should have experience not only with programming concepts but also with Python specifically, so they should be encouraged to review the previous lesson to keep their Python knowledge fresh prior to this lesson being taught
  - Student groups now be split up based on individual confidence levels in programming, which may be tracked with a survey done after the first few programming lessons
- As the main assignment of this lesson, students will create a new Python project, and develop two methods
  - Alongside a packet of information to guide students through the development of this program, the teacher will also provide a CSV file to students for use in one of their methods
  - First, students will create a method that uses an equation for gravitational force between two particles of specified mass at a specified distance in order to plot gravitational force as a function of distance
    - Teacher will give students these equations once again at the beginning of class so that students use it correctly
  - Second, students will create a method where they import data from a CSV file, similarly to as in the first Python lesson, and use the data to create a plot for electrostatic force vs distance
- Once students have debugged and properly compiled their methods, they should call the methods to output two graphs, one of which will be gravitational force vs distance, and the other will be electrostatic force vs distance



### Take-Home Tasks:

- As the final portion of this lesson, students will create a write-up comparing and contrasting the two forces and their graphs.
  - If this is not completed during class time, this can be a take-home task
  - This writeup should specifically talk about how the graphs show how the electrostatic force is parabolic, and is therefore related to the square of the distance between charges in a manner similar to that of the force between two celestial bodies
  - For bonus points, students may research, program, and discuss how these forces would act on the same objects
    - Would the electrostatic force be greater than the gravitational force? Or vice-versa? How do you know this is true logically?

## DISCUSSION

### *Field Testing*

Testing the efficacy of these lessons was a two-part process. Firstly, two rounds of classroom testing were conducted at two separate high schools. The first round was at Milford High School, in collaboration with Avery Travis, and the second at Nashoba High School, in collaboration with Robert Gendron. Across the two schools, 44 students in total worked through the first lesson. Secondly, the testing process was enriched through collaborative efforts with experienced high school educators. Through several conversations and written reviews, these teachers contributed a myriad of anecdotal evidence highlighting the strengths and weaknesses of the lesson designs.

During the first round of testing with Ms. Travis, two major areas of improvement were noted. The application of a partner programming method called the “Driver/Navigator method” established a group dynamic that kept a large majority of the students engaged throughout the lesson. As the students worked through the sections of the lesson activity, they would alternate between being the driver, the person typing things into the computer, and the navigator, who would inform the driver on what to type. Later in the lesson activity, the inclusion of a velocity column in part 2 sparked an intriguing teaching opportunity. When dragging a formula down in Google Sheets, any referenced cell locations will adjust for each filled cell. As a result, the distance output based on a formula using a changing initial velocity value would be incorrect after  $t = 0$ . This

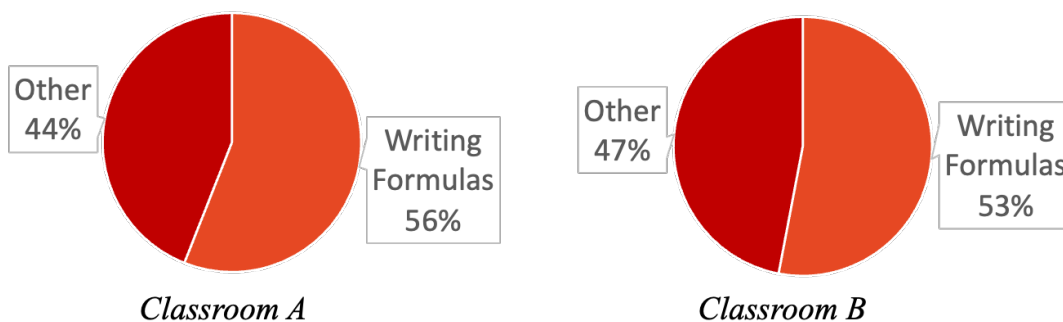
would prompt students to reevaluate their formula and would open the door to introducing static cell references in an organic way.

Following the first lesson, another major takeaway came from the student reflections. The most common point of contention for the students was making formulas out of the kinematics equations. In fact, 56% of non-empty student reflections mentioned this. This may have been for one of two reasons: the lesson ineffectively introduced Google Sheets syntax for writing formulas, or the students were simply inexperienced with the kinematics equations themselves. Potential solutions would be to either dedicate intentional time before the lesson activity to familiarize students with mathematical operations in spreadsheets, or to make sure students have a working knowledge of the physics before this lesson is even taught. As the second round of testing was done with a group of students who had much more experience with kinematics, I was able to incorporate both of these solutions.

During the second round of testing, Mr. Gendron spent time at the beginning of class going over the spreadsheet-equivalents for basic mathematical operations, namely addition, subtraction, multiplication, division, and exponentiation. Unfortunately, about 53% of the student reflections still mentioned setting up equations as the main struggle. While the percentage of students that specifically cited this decreased slightly, the proposed solutions were clearly unsuccessful. Another aspect of the student reflections, however, pointed to a different solution.

Many students mentioned that they believed continued practice with the concepts proposed in this lesson would be necessary. As a result, the issue with writing equations may simply be due to it being a completely new experience rather than a lack of physics / syntactic knowledge. This brings up an interesting issue, which is that the issue with creating formulas in spreadsheets may be more psychological than cognitive. New experiences, even when they are grounded in

seemingly understood concepts, seem to pose difficulties. A potential solution to this would be to incorporate some formula writing practice in an informal setting at the beginning of the lesson. For example, the instructor could have students write out each of the kinematics equations with spreadsheet syntax, then rewrite them with cell references once the given values have been written in a sheet. This way, the act of writing the formula becomes more of a process, where students can see where each part of the formula comes from all the way back to the kinematics equations as they initially learned them.



*Figure 2: Common Areas of Struggle (Lesson 1 - Kinematics)*

Between the two rounds of in-classroom testing, I realized the importance of continued support and repetition. The lessons were designed to be completed in single class periods, but this was based on the assumption that a classroom could go straight through each part of each activity continuously. I did not anticipate the slower pace exhibited in both classrooms as a result of repeating statements and going over concepts multiple times. This could be addressed by simply splitting the lessons up a bit more, perhaps having the activities as the final part of a multi-period

block. Perhaps, by emphasizing a greater conceptual foundation, there would be less issues caused by the novelty of working with spreadsheets and/or Python.

### *Collaborative Efforts*

Due to time constraints, more than two rounds of field testing was not possible. As a result, emphasis was shifted to collaborating with high school teachers to refine the pedagogy of the lessons. In these collaborations, three educators received the lesson plans, scaffolded instruction outlines, student materials, ideal activity submissions, and topical presentations. With these, Maria de Vuono-Homberg, Thomas Young, and Ben Petkie all provided both positive reinforcement and constructive criticism based on their extensive experiences in real-world teaching environments.

The collaborative efforts particularly emphasized the importance of scaffolding and support for both the teacher and the students. The complex process associated with teaching interdisciplinary lessons can be difficult for both parties, so providing supportive resources is essential for the success of the lessons. Some suggestions for how to expand and improve on these resources include more explicit walk-throughs of certain skills, supplementary visual aids, and instructional videos. Improved contextualization opportunities could also diminish any knowledge disparities that exist in different classrooms. For example, combining intermediate skill development activities such as coding in blocks with real-world physics situations such as sports would improve the long-term efficacy of the lessons. This particular point exposed another potential improvement, which is the inclusion of greater coding support.

Between English language learners, finite attention spans, and limited time frames, the amount of reading provided in the lessons could present a major barrier as students jump from

spreadsheets to coding. A major point emphasized by the three teachers I spoke with was that experiential learning, especially at the high school level, tends to be more effective than simply reading through material. While the lessons as a whole build on one another, the intermediary skills within the lessons could have more emphasis. This is especially relevant in Lesson 3 and 4, as coding will almost always be an entirely new experience, skill, and world for students to work through. As a result, the later lessons would benefit from more skeletal code blocks, more interactive coding activities, and generally a slower development of skills with less reading.

## **FUTURE STUDIES**

As in many fields, Artificial Intelligence stands as a transformative technology that will inevitably be used in education. In the context of this project, which aims to provide computational physics lessons to combine physics and computer science, AI holds incredible promise. As mentioned previously, throwing students and their teachers into interdisciplinary lesson objectives can cause complications. Often, the teachers themselves may have minimal coding experience themselves. As large language models continue to progress in their capabilities, especially within computer science, AI could definitely aid with debugging of low-level code. For students, AI could be implemented as a strategic scaffolding tool as they learn the basics of computer science, and for teachers that may not have much coding experience it could help them spot errors and provide solutions back to the students.

The decision to build the lesson activities around group work was, although supported by various literature, somewhat arbitrary. In fact, arguments have at times been made against using group work as a teaching method. As a student in a classroom at Milford High stated, learning

computer science is like learning another language, and we don't learn the ABCs through group work. While my decision to incorporate group work as a major aspect of the lessons was based on a desire to engage students and encourage social collaboration at a pivotal age, there are other benefits to individual work that could add more to the lessons. As a result, it would be interesting to see research comparing the two in physics and computer science directly. Studies into this particular topic may provide some clarity into which would better encourage classroom-wide learning.

In addition to exploring the potential for Artificial Intelligence integration and comparing individual / group work dynamics, the application of problem-solving strategies as analogies for programming frameworks presents an interesting theme for future studies. Problem solving templates like G.U.E.S.S. (given unknowns equations substitutions solve) are often referenced in high school physics as a support mechanism for students, and can be easily adapted for developing computer science skills. By drawing parallels between these types of approaches and programming methods, teachers can highlight abstract coding concepts through familiar processes. G.U.E.S.S. for example can be analogous to the declaration or instantiation of variables and equations. Another example is the iterative nature of the scientific method, which can be analogous to the process of debugging and improving code. Research into how these analogous processes could be exploited would have a major impact on the landscape of computational physics teaching methods.





## Appendix A – Complete Lesson Plans

### *Lesson Plan 1 - Intro to Spreadsheets*

#### Massachusetts HS Physics Standard:

HS-PS2-10 (MA): Use free-body force diagrams, algebraic expressions, and Newton's laws of motion to predict changes to velocity and acceleration for an object moving in one dimension in various situations.

#### Learning Objectives:

##### 1. Physics

- a. Students will be able to communicate an understanding of velocity, speed, acceleration, mass, inertia, weight, displacement (change in position), and distance.

##### 2. CS

- a. Students will be able to identify the main aspects of the google sheets window.
- b. Students will be able to use the basic cell-based functions available in google sheets, including insert, delete, drag, highlight, fill, etc.
- c. Students will be able to write arithmetic formulas in google sheets that utilize built-in commands in order to do basic math operations.

- d. Students will be able to identify what graphs are best suited for different types of data sets, as well as understand how to create these different graphs using given tables.
- e. Students will understand and know when to use conditional formatting (filter) as well as conditional formulas (IF statements)

Summary of Tasks / Actions:

- Hook activity: Split class into small groups and give them the following questions to discuss
  - Have you used google sheets in the past? If so, what have you used it for?
  - How could google sheets be helpful to physics researchers?
- The first section of this lesson will be a guided lecture
  - You will go over the following aspects of the google sheets window to begin:
    1. Creating a file
    2. Clipboard (cut/copy/paste)
    3. Font tools (color, size, style, etc.)
    4. Alignment Tools (wrap text, merge, etc.)
    5. Insert tab (tables, pictures, charts, graphs)
    6. Arithmetic operators (reference cells/groups of cells)
    7. Drag-down feature
  - Students can follow along with all of these using personal devices

- Students can also take notes on these functions, as any of them could prove useful in future uses of google sheets software
- The second section of this lesson will be an activity to reinforce the above skills - Basic Kinematics Problem
  - Give students a 2-D motion scenario (such as a ball getting dropped off a cliff)
  - Given values
    - Acceleration due to gravity
    - Time it takes for ball to hit the ground
    - Ball is dropped from rest
  - Part 1 - Put values into google sheets and write a formula where you reference the correct cells to calculate the distance the ball falls off the cliff
  - Part 2 - Create a new sheet and use your formula again but create a table of time values from 0 to given time in intervals of 0.1 seconds and how far the ball has fallen at each time. Use the drag-down feature to complete this. Then create a graph based on this table that represents displacement vs time with all of your data points shown on the graph. Export and save the graph as a separate file and submit both files to complete assignment.

### Take-home tasks:

The take-home work after this lesson will be preparing for the google sheets lab activity that will augment the Kinematics and 2-D motion portion of the physics class. This take home assignment will essentially be a pre-lab review of both physics concepts and the essential google sheets skills

that will be applied in the lab. While everything covered in this lesson is essential to having a foundational knowledge of google sheets, there are a few specific features that are specifically beneficial when working with physics, which include creating graphs, editing graphs, arithmetic operations, and drag-to-fill. These will be used extensively in the lab activity, and a solid understanding of these particular topics will be very important for collaborative, student-based learning to be successful in the following lesson.

## *Lesson Plan 2 - Kinematics and 2-D motion in Google Sheets*

### Learning Objectives:

#### 1. Physics

- a. Analyze and compare the motion of objects using position/distance vs. time, velocity/speed vs. time, and acceleration vs. time graphs.

#### 2. CS

- a. Students will be able to manipulate tables using arithmetic functions and drag-to-fill.
- b. Students will be able to create graphs from tables with correct axes labels and titles.
- c. Students will be able to add relevant trend lines based on the scenarios provided by different data tables.
- d. Students will be able to customize graphs in google sheets using advanced formatting features.
- e. Students will be able to export graphs from google sheets to be used in separate documents.

### Summary of Tasks / Activities:

- Hook Activity: Projectile Motion
  - Get a volunteer and throw a ball in the air across a distance
  - Then throw a ball straight up and down

- Ask students what the difference is
  - Based on responses, mention the idea of horizontal velocity having no acceleration, and vertical motion following the same path in both examples
- Demonstrate the parallel with one final demonstration, but rather than thrown the ball at an angle, like in traditional projectile motion problems, slide across the floor in a chair and toss the ball straight up and down
  - The ball will follow a projectile path, even though you seem to just throw it perpendicular to the ground
- Give students packet to follow
  - Provide link to position time chart in both x and y directions
  - Students are tasked with taking this chart and creating velocity-time and acceleration-time charts for both directions in separate sheets
  - Students will then graph each chart and customize them however they would like
    - Necessary graph details
      - Labels
      - Units
      - Title
      - Correct axes
      - Correct graphs
- Teacher will mostly just monitor the progress of different groups
- Groups will be split up based on previous knowledge on google sheets

- Groups will be 2-3 people at most, and while they will be working together to get through the parts of this activity, each student will be following along on their personal devices
- Each student will submit their own work

Take-home tasks:

Since Google Sheets is available on any personal device, students would be expected to complete any part of this activity at home that they did not get to in class. As this lesson is collaboration-based, this would encourage students to communicate outside of class and establish a group dynamic that emulates what it will be like in college.

### *Lesson Plan 3 - Simple Harmonic Motion in Python*

#### Massachusetts High School Physics Standard:

HS-PS2-10(MA). Use free-body force diagrams, algebraic expressions, and Newton's laws of motion to predict changes to velocity and acceleration for an object moving in one dimension in various situations.

#### Learning Objectives:

1. Physics
  - a. Students will understand the principles of Simple Harmonic Motion and apply them to graph aspects of spring-based motion.
2. CS
  - a. Students will be able to create a .py file, save it, compile it, and run it.
  - b. Students will be able to create graphs in python using built-in plotting functions.
  - c. Students will be able to properly label graphs in python.
  - d. Students will be able to export graphs for use in external files.

#### Summary of Tasks / Activities:

- Before class: packet to follow as an introduction to python IDE



- Basic window features only, as well as how compiling and running a program works
- Functionality of python as a programming language will be described mainly as a parallel to what was taught in google sheets
  - Variable declaration and data types
  - Arithmetic functions
  - Function calls
  - Comments
- “Hello, World!” program
- Hook Activity: Model a Spring Systems
  - Teacher will begin this lesson showing students a system where a block is oscillating on a frictionless surface while attached to a spring with a known spring constant
  - Students will be tasked to model this system and the different types of energy associated with it
    - **\*\*this computational lesson should be following a broader introduction to SHM, including an overview of the position, velocity, and acceleration of an oscillating system\*\***
  - Models will be a springboard to create a set of energy equations to describe this system
- With a base of knowledge in programming concepts from google sheets, and the basics of Python from the pre-class review, students will be tasked with creating a basic program that simulates an SHM system and create graphs for several components

- As in the previous google sheets activity, students will be split into groups to complete this project
- Students will be given values for mass, spring constant, and amplitude
- With these values they should write a program that uses the SHM equations established at the beginning of class to create graphs for kinetic energy, potential energy, and total energy

## *Lesson Plan 4 - Electrostatic and Gravitational Forces*

### Massachusetts High School Physics Standard:

HS-PS2-4. Use mathematical representations of Newton's law of gravitation and Coulomb's law to both qualitatively and quantitatively describe and predict the effects of gravitational and electrostatic forces between objects.

### Learning Objectives:

#### 1. Physics

- a. Students will be able to analyze and compare systems of two objects with different charges and distances apart to compare the magnitude of the electrostatic forces.
- b. Students will be able to analyze and compare systems of two objects with different masses and distances apart to compare the magnitude of the gravitational forces.
- c. Students will be able to compare and contrast the graphs of gravitational force and electrostatic force to better understand the forces of our universe.

#### 2. CS

- a. Students will be able to develop multiple methods to provide functional procedures in their code.
- b. Students will understand what object-oriented programming is, and be able to explain how the development of methods is a part of this particular strategy.
- c. Students will be able to program loops to calculate data points for their plots.

- d. Students will compare and contrast the graphs of gravitational force and electrostatic force

Summary of Tasks / Activities:

- Hook Activity:
- Similarly to in the second google sheets lesson, the teacher will act more as a supervisor over student groups
  - This lesson should be implemented at the start of the Electricity and Magnetism unit
  - Students at this point should have experience not only with programming concepts but also with python specifically, so they should be encouraged to review the previous lesson to keep their python knowledge fresh prior to this lesson being taught
  - Student groups now be split up based on individual confidence levels in programming, which may be tracked with a survey done after the first few programming lessons
- As the main assignment of this lesson, students will create a new python project, and develop two methods
  - Alongside a packet of information to guide students through the development of this program, the teacher will also provide a CSV file to students for use in one of their methods

- First, students will create a method that uses an equation for gravitational force between two particles of specified mass at a specified distance in order to plot gravitational force as a function of distance
  - Teacher will give students these equations once again at the beginning of class so that students use it correctly
- Second, students will create a method where they import data from a CSV file, similarly to as in the first python lesson, and use the data to create a plot for electrostatic force vs distance
- Once students have debugged and properly compiled their methods, they should call the methods to output two graphs, one of which will be gravitational force vs distance, and the other will be electrostatic force vs distance

#### Take-Home Tasks:

- As the final portion of this lesson, students will create a write-up comparing and contrasting the two forces and their graphs.
  - If this is not completed during class time, this can be a take-home task
  - This writeup should specifically talk about how the graphs show how the electrostatic force is parabolic, and is therefore related to the square of the distance between charges in a manner similar to that of the force between two celestial bodies
  - For bonus points, students may research, program, and discuss how these forces would act on the same objects

- Would the electrostatic force be greater than the gravitational force? Or vice-versa? How do you know this is true logically?

## Appendix B – Student Packets

### *Pre-Class Review: Intro to Spreadsheets*

Spreadsheets are an essential tool in many different settings, and knowing how to use them is a very useful tool. Software such as Excel and Google Sheets can be used to create and edit tables, design professional-grade graphs, and even do complex mathematics. As a result, spreadsheets are used every single day in many different industries. While filling in tables and creating plots may seem pretty straightforward, working with spreadsheets actually uses a lot of the same concepts as programming! In fact, excel and google sheets were created to be used as programming platforms only with minimal coding. For this reason, understanding the basics of spreadsheets is the first step in learning programming for this course.

In tomorrow's class, I will be walking you through the basic features of Google Sheets on the projector, and you will be following along on your personal device (Chromebook / laptop). Afterwards, you will be asked to work in small groups to create a Sheets file, write formulas to create projectile motion data points, and design a graph representing this type of motion. Below are a couple of short answer questions that should be completed before class... but don't worry, there are no right or wrong answers! They are just meant to get you thinking about spreadsheets, and you will get the chance to discuss your responses with your classmates at the beginning of class. Feel free to use the internet to help you answer these questions if you would like, being able to conduct research is another valuable skill for you all to develop :) Please use the space on the next page to write down your responses!

1. Have you used Excel / Google Sheets in the past? If so, what have you used it for?
2. How could Excel / Google Sheets be helpful in physics or any other industries?



*Intro to Spreadsheets: Projectile Motion Activity*

Part 1)

A baseball initially at rest is dropped off a high cliff. It takes 5 seconds to hit the ground at the bottom, and you can take acceleration due to gravity to be equal to  $9.8 \text{ m/s}^2$ . Assuming negligible air resistance, how tall is the cliff? In order to solve this problem, you will be using Google Sheets. To format this correctly, you should fill column A with variable names, and column B with the corresponding values. The first few rows should include the given values, but the last number in column B should be the correct height value. For full credit, write a formula to solve for this height using the built in arithmetic operations we reviewed.

Part 2)

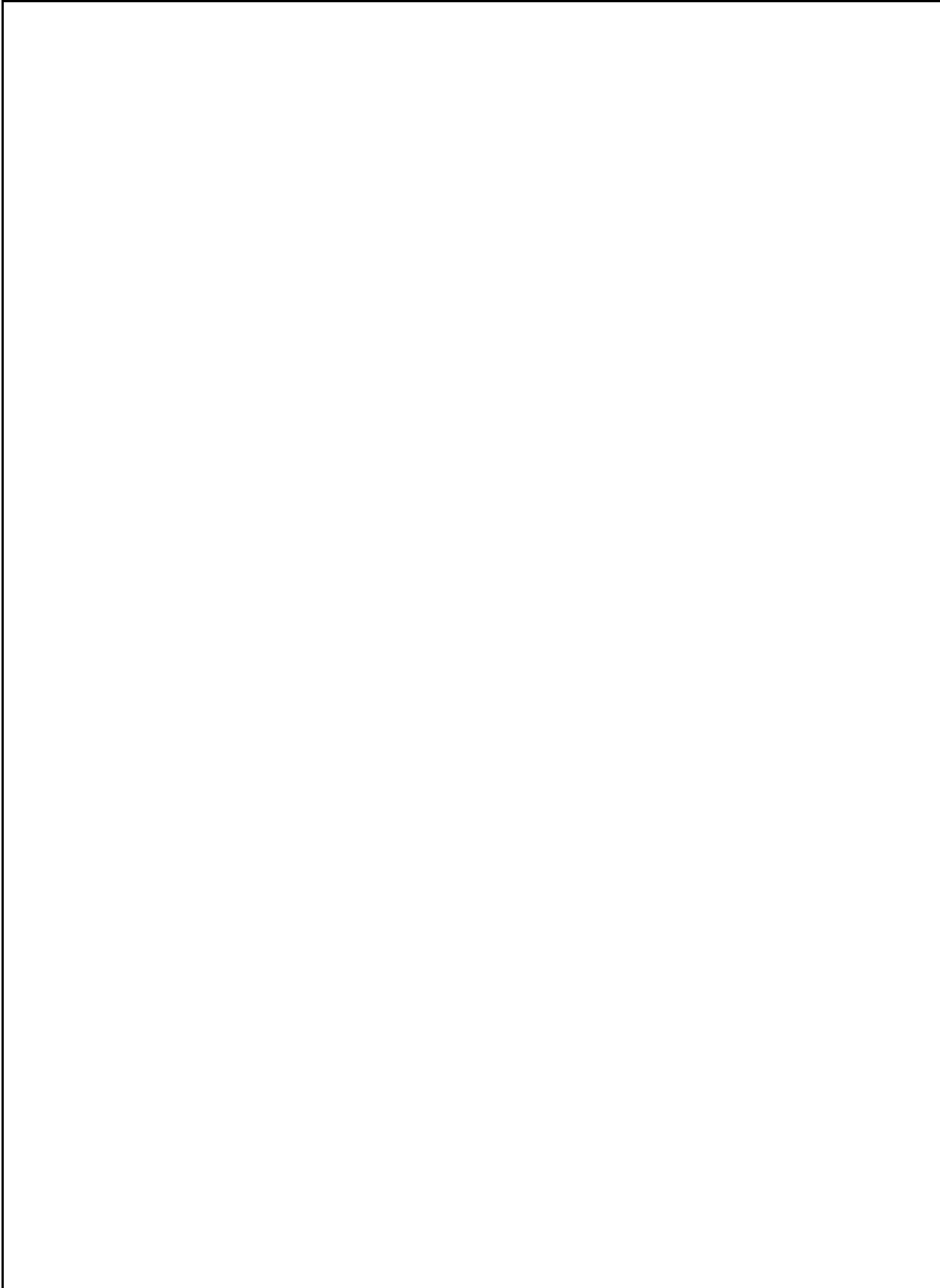
For this part of the activity, you will be creating a table of time and position values and a graph representing the ball's motion based on the values provided above. In column A, you should write in time values from 0 to 5 seconds in 0.1 second intervals. In column B, you should use a similar formula to the one you created in part 1 to find position values corresponding to each time value. As you no longer have cells containing the given values, you should now write the values for each in your formula directly. Remember, there are a lot of helpful features built into Google Sheets that should make the process of creating your table quite fast! Once you have filled in your table, you should plot your data by inserting a graph. For full credit, make sure you format your graph correctly with a title and axis labels.

### *Take-Home Assignment*

If you were not able to complete the activity by the end of class, you can continue working with your group outside of class to finish making your graphs. Remember, while you are encouraged to work together, each of you should make your own submissions for credit. Once you have submitted your graph and spreadsheet file, you will complete an individual reflection below about your experience with the activity.

1. What was your level of knowledge of spreadsheets before this lesson?
2. Do you feel confident in your ability to use basic spreadsheet features after the activity in class?
3. What part of the assignment did you / your group struggle with the most?

This homework will not be graded, so be as honest about your experience as possible! Use the space on the next page to write your reflection.



## **Kinematics and 2-D Motion with Spreadsheets**

### Part 1)

For today's activity, you all will be using Google Sheets once again to develop comprehensive charts representing projectile motion and create graphs to go along with them. To start off, copy and paste the data points from the link on the screen into a new Google Sheets file. I have provided you with a set of time values, x-position values, and y-position values. Similarly to the last lesson in spreadsheets, you will be utilizing built-in arithmetic operations to manipulate your chart and fill in four more columns of data: x-velocity, y-velocity, x-acceleration, and y-acceleration. Once you have successfully filled your chart (which should now have 7 total columns), make sure all of your values actually make sense by comparing with what we discussed at the beginning of class. If everything looks good, you can move on to the next part of the assignment!

### Part 2)

For this part of the activity, you will have the opportunity to be a bit creative... Now that you have completed your data chart, you will be moving each set of data (along with the corresponding time values) into a separate sheet. Then, create graphs for each one. In total, you should have six graphs by the end of this. Google Sheets has a lot of available features, so feel free to customize your graphs as much or as little as you would like. The only parameters for this part of the assignment is that you include the following features in your graphs: a title, correct axes, axis labels, and units. Other than that, get creative!

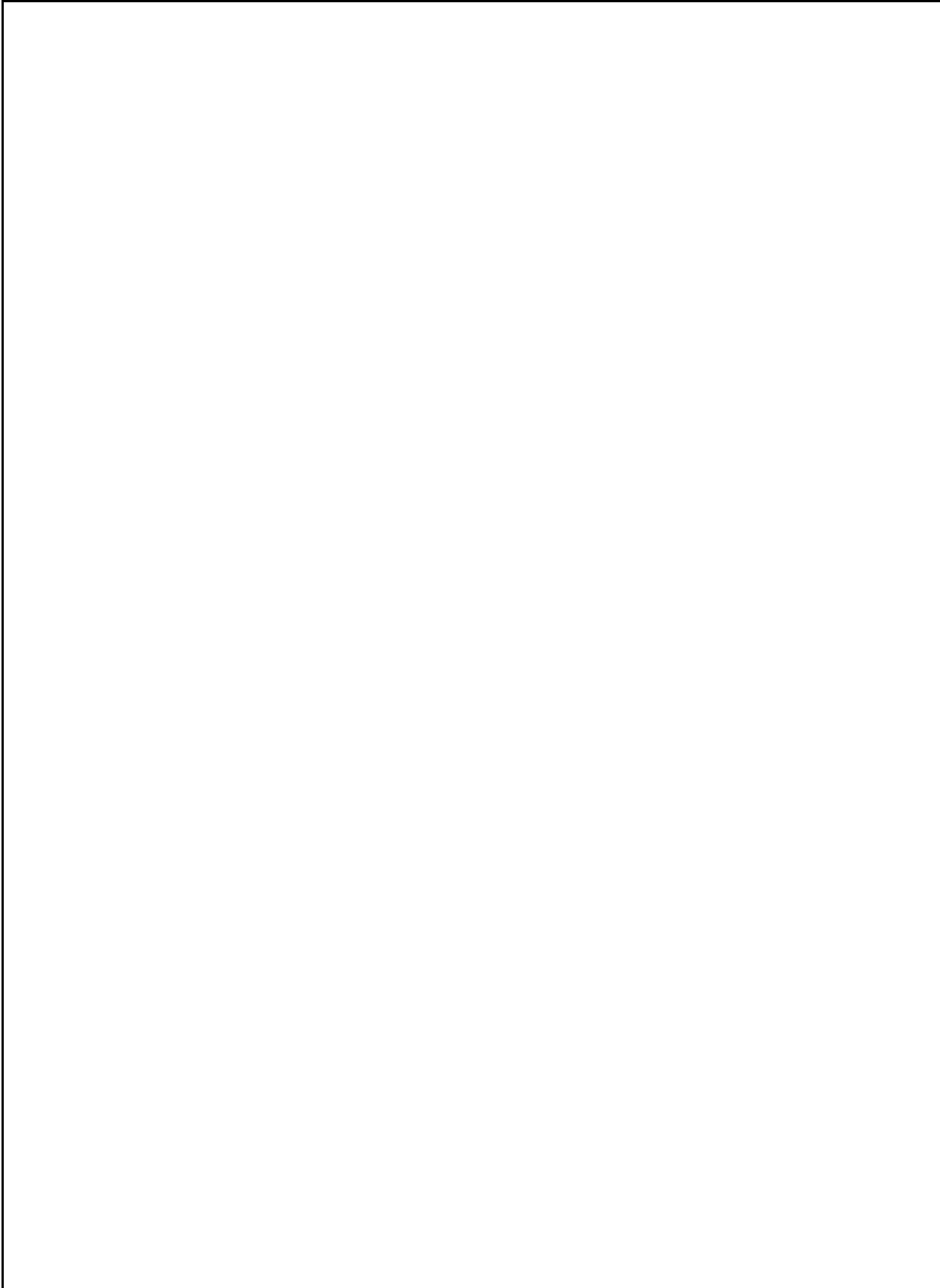
## **TAKE-HOME ASSIGNMENT**

If you were not able to complete the activity by the end of class, you can continue working with your group outside of class to finish making your graphs. Remember, while you are encouraged to

work together, each of you should make your own submissions for credit. Once you have submitted your graph and spreadsheet file, you will complete an individual reflection below about your experience with the activity.

1. What was your level of knowledge of spreadsheets before this lesson?
2. Do you feel confident in your ability to manipulate cells and create graphs after the activity in class?
3. What part of the assignment did you / your group struggle with the most?

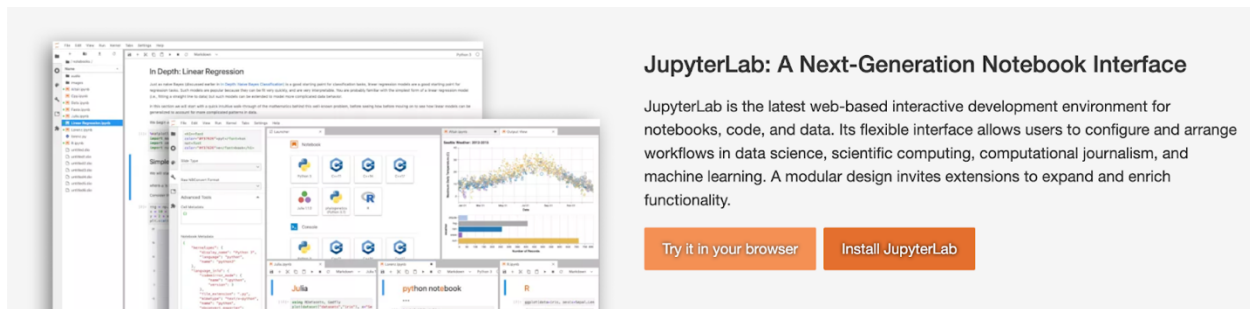
This homework will not be graded, so be as honest about your experience as possible! Use the space on the next page to write your reflection.



## Pre-Class Prep

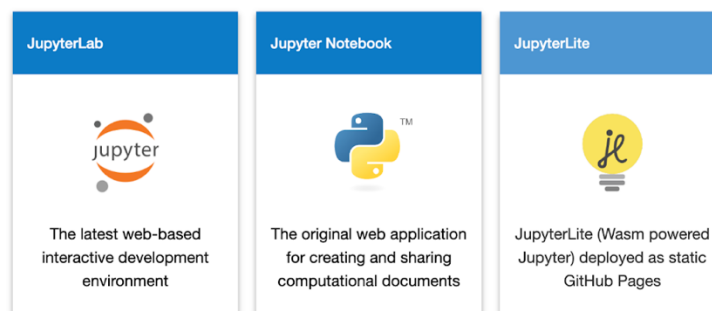
### Step 1) Creating a Trinket Account

1. Enter the following link into your browser: <https://jupyter.org>
2. Scroll down and click on “Try it in your browser,” in the JupyterLab section:



3. You should now see a list of different applications that are available through JupyterLab.

We will be using JupyterLite as it provides a sleek and easy-to-learn user experience.



4. Once you have clicked on JupyterLite, you will be brought to a launcher screen where you can choose what type of file you would like to create. At this point, you should click on the Python file in the second section marked as “Console.” This will open a python file that you can edit and run. In JupyterLite, programming is done in cells, which allows you to edit or run your program in blocks. You can run the code in your cell by clicking on “run >> run cell” in the top left of your screen. Instead of running everything all at once, separating pieces of code into cells allows

programmers to test small tasks much more easily. For the sake of simplicity, you will be writing all of your code in one cell and running the cell as a complete program. In order to save your code and ensure it will not be lost the next time you enter JupyterLite, make sure to copy and paste your code into a Python File and save it. You can do this by clicking “File >> new >> Python File”.

5. As an introduction to Python, you will simply be writing a program that prints out a statement. Rewrite the following line of code into your program and press run (if you copy and paste, the formatting will not be correct and you may run into an error):

```
print(“Hello, World!”)
```

Congratulations! You have written your first line of code, and your first program! Next, you will be going over some of the basic capabilities of python programming so you are well-equipped for the in-class activity tomorrow :)

### **Programming 101... kind of**

You may not believe it at first, but you already know a good bit about programming! In fact, you have already learned about some of the most important principles of programming just from working with spreadsheets. Using the drag down function in Google Sheets is essentially equivalent to using a loop. Using built-in operations to write formulas in spreadsheet cells is analogous to making function calls in Python! Even using titles in your charts (position, velocity, time, etc.) is just like creating variables. Although it may not have felt like it, you have been coding for a while now! To build on your existing knowledge of programming from spreadsheets, I will be going over six main concepts in a bit more depth so you will feel more comfortable with our in-class activity. The six concepts I will be reviewing are variables, data types, functions, loops,



and comments. As I go through each of these, I encourage you to think about how they relate to aspects of spreadsheets you used in the previous lessons.

## **VARIABLES**

Variables are used to store data values using a particular label / name. When you create a variable, it is conventional to give names to your variables based on what they will be used for. It is also convention to start your variable names with a lowercase letter. An important aspect of Python is that it is a case-sensitive language, so the case of each letter in a variable name matters. In other words, `aVariable` and `avARIABLE` would be treated as two completely different things by your computer. Whenever you reference your variables, make sure to spell them correctly and use all the correct cases for each letter. Variable names can consist of letters, numbers, and underscores, and they cannot have any spaces. They **MUST** start with a letter or underscore. If you begin a variable name with a number, the computer will give you an error when you run your code. Rules like this are inherent to the Python programming language and are part of what is called the syntax. Syntax is essentially the grammar of the language. Just as the English language has rules, so too do programming languages. The last key aspect about variables is that you cannot assign a variable a name if it is a reserved keyword. For example, your variables cannot be named “if,” “while,” or “for.” Here is the syntax for creating a variable:

```
variable_Name = value
```

## DATA TYPES

Data types are classifications for your variables that specify to your computer what type of data it is working with. Below are some examples of data types you may use in your code:

String - A *String* is a line of characters between a set of quotations. Anything in your code that is between quotations is treated as a string. The shortest possible string is the empty string, which is specified as `""`. No spaces, characters, symbols, nothing. A string can be as long as you would like, as long as you don't close the quotation!

Char - A *char* represents a single character (a letter number or special character) between a set of single-quotations.

Integer - An *int* is the most basic of all the numerical data types, and it denotes any integer value. This can be negative, positive, or 0. Just as you may have learned in mathematics, an integer does not have any decimal.

Float - A *float* variable is another numerical data type, except this time it takes decimal values. In computer science, the decimal point is sometimes called the "floating point," which is where the float data type gets its name!

Boolean - A *boolean* variable is given one of two variables: TRUE or FALSE. Boolean values are often used for logic-based code, something like "if \_\_\_\_\_ is TRUE, then ..."

An interesting aspect of python is that it is a dynamic programming language, meaning you don't need to specify data types when creating variables. In most programming languages, you would

need to do so. As a result, data types are very important to understand so you can keep track of what types of data you are working with. Fortunately for you, when you are working in python the computer will make assumptions as to what data type your variables are based on the values you give them.

## **FUNCTIONS AND FUNCTION CALLS**

Just like in Google Sheets, Python provides a vast library of built-in functions. Functions are basically back-end programs that, when referenced in your own code, allow you to do essential tasks in a single line. Every function has two main aspects: Input and Output. For input, all functions have certain parameters. Parameters are the variable arguments you enter as input that are necessary for the function to do its job. Every function outputs a variable of a particular data type. It is important to keep track of these data types so you don't run into issues trying to do something with a function's output that isn't allowed (it outputs a string and you try to use it in an arithmetic problem). An example of a function is *pow()*. The *pow()* function can be used to raise a number to a particular exponent. As you can probably guess, *pow()* needs two input values: the base and the exponent. In return, *pow()* will output a number value (this particular function can work with either integers or floats). All function calls have the following syntax: `functionName(parameter1,..., parameterN)`. Below is what a function call to *pow()* might look like.

```
base = 2      # assign a value to a variable base
exp = 8       # assign a value to a variable exp
exponentNum = pow(base,exp)      # exponentNum = 256
```

## LOOPS

Loops are coding structures that can be used to repeat a block of code a certain number of times. In other words, when you write a loop into your code, your computer will follow the same set of instructions over and over until a particular condition is met. For example, if you wanted to print something out 10 times, you might use a loop that iterates 10 times and each time prints whatever you want. When this type of structure is read by the computer, it will not move on to the next part of the code until it stops looping. As a result, you should be careful to not create an infinite loop, as your computer will get stuck and your program will not run properly. There are two main types of loops that you can use: a *for* loop and a *while* loop.

A *for* loop is used when you know how many times you would like to repeat your set of instructions. Here is the syntax of a *for* loop that prints out number from 1-9:

```
for aNum in range(1,10):    # aNum goes up by 1 in each iteration
    print(aNum)            # in each iteration, aNum gets printed
```

A *while* loop is used when you do not necessarily know how many times the loop has to repeat, but you have a particular condition that is the indicator for when the loop will end. Here is the syntax for a *while* loop that restarts a game over and over until the player says they don't want to play again:

```
while playerResponse == TRUE:    # loops until boolean playerResponse is no longer TRUE
    newGame()                    # creates a new game in each loop iteration
```

## COMMENTS

One of the most important aspects of programming is keeping detailed notes within your code. In the world of computer science, code is often passed around and edited by a number of programmers. As a result, annotating your code so others can more easily read it is essential. The way programmers do this is by using comments. Comments are essentially blocks of text written after a pound symbol that are ignored by the computer when it goes through your code. You can write comments exactly the way you would write in any text file. I have used comments throughout the above instructions to explain what different lines of code mean, and it is used in the sample code you will receive for the in-class assignment to separate different sections of your program. Writing detailed comments in your code is a valuable skill, so I hope you use the in-class assignment to develop this skill!

### **ADDITIONAL POINTS**

1. '=' and '==' are different things. '=' is used when assigning a value to a variable. It is referred to in programming as the *assignment operator*. '==' on the other hand, is used to check for equality between two values. It is referred to as the *equality operator*.
2. Arithmetic is very straightforward in python. Just as when you learned it in math, you must follow the order of operations. You can use '+' for addition, '-' for subtraction, '\*' for multiplication, '/' for division, and '^' for exponentials. Any more advanced arithmetic operations, such as trigonometric functions (sin, cos, tan), will be explained in class.
3. Indentations are necessary in blocks of code... basically, when you use things like loops, anything that is supposed to happen inside the loop has to be indented in your code. You

can simply use the tab button to achieve this. Make sure not to add any spaces though, this will mess up the computer.

4. Don't forget colons when writing loops! They are used by the computer along with indentations to signal a block, or section, of code.

## Simple Harmonic Motion in Python

Part 1)

For this activity, you will be creating a graphical representation of the energies associated with simple harmonic motion of an object attached to a spring. Assume that the object is oscillating on a frictionless surface, so it is not being affected by any external forces. Your final graph should show how the potential, kinetic, and total energy of the system change over one full period. Based on the given values, you will be tasked with writing equations for angular frequency, period, kinetic energy, potential energy, and total energy. In order to do so, you will have to utilize some of Python's built-in capabilities. The Python library of functions also includes sub-functions, which are basically small-scale functions for more specific tasks. For example, Python has a `math` () function that has a number of mathematics related sub-functions that can be called. If you wanted to generate a random number, you could call `math.random()`. Notice the syntax for calling a sub-function is to add a period between the function and sub-function names. In this first part of the assignment, you will have to use the `sin` and `cos` sub-functions to create your energy equations. The syntax for using these trigonometric sub-functions is the following:

```
math.sin(theta_value), math.cos(theta_value)
```

Once you have developed your energy equations, you can move on to part 2.

## Part 2)

The next part of this assignment is to plot your potential, kinetic, and total energy equations on a single graph using python. Now that you have developed equations for each type of energy as a function of time, you can use some more sub-functions to create your graph. In the sample code I have provided, the function for creating graphs in python is called plt. In order to create your graphs, you should make use of the following:

1. `plot(x, y)` - creates a line plot using parameters `x` and `y` for all point values. For the sake of this project, you can use the `x_values` list you previously created for the first parameter, and an energy equation for the second parameter. To create multiple lines on your graph, you can call this sub-function multiple times, once with each energy equation.
2. `title(aString)` - assigns `aString` as the title of your graph. The two ways you could use this function are by creating a String variable and referencing it when you call the sub-function (ex: `title(myTitle)`), or you can simply write the String you want in the sub-function (ex: `title("this is a title")`).
3. `xlabel(aString)` and `ylabel(aString)`- works similarly to `title`.

You can use the code template on the next page to get started. Good luck!

### *Starting Code*

```
import matplotlib.pyplot as plt      #imports built-in library for making graphs
import numpy as np                  #imports built-in library for doing advanced computing
import math                         #will be used for math.sin() and math.cos()

# given values
```

```
m = 2
amplitude = 0.2
k = 10

#variables to create: angular frequency (w) + period (T)

# part 1: creating energy equations: kinetic, potential, total
time_values = np.linspace(0, period, 100) #list of 100 values from 0 to a period length

# part 2: making graphs

# output graphs
plt.show()
```

### **TAKE-HOME ASSIGNMENT**

If you were not able to complete the activity by the end of class, you can continue working with your group outside of class to finish making your graphs. Remember, while you are encouraged to work together, each of you should make your own submissions for credit. Once you have submitted your graph and spreadsheet file, you will complete an individual reflection below about your experience with the activity.

1. What was your level of knowledge of programming before this lesson?
2. Do you feel confident in your ability to write code and create graphs in python after the activity in class?



3. What part of the assignment did you / your group struggle with the most?

This homework will not be graded, so be as honest about your experience as possible! Use the space on the next page to write your reflection.



## Pre-Class Prep

In the last lesson, you worked with several sub-functions and functions. While those sub-functions were built-in, you can also write sub-functions on your own code. In the world of computer science, these sub-functions are more universally referred to as *methods*. Methods are blocks of code that allow you to write self-contained instructions for specific tasks. Methods provide many important roles to programmers. One role is providing modularity, meaning that methods split up your code into practical chunks that each handle individual jobs. Another role is creating reusable code, as methods can be called as many times their job is needed. To create a method, you must use the keyword 'def,' followed by the name of your method, a set of parentheses, and a colon. The block of code that belongs to a particular method should all be indented. Below is an example of method creation:

```
def my_method():  
    #code for my method  
    #must all be indented
```

In order to actually use your method, you must call it explicitly. Similarly to calling built-in functions and sub-functions, the syntax for calling a method is the following:

```
methodName(parameter1,..., parameterN);
```

## Electrostatic and Gravitational Force

Part

1)

For this assignment, you will be creating two methods in Python. The first will output a graph of

gravitational force vs position between two objects, and the second will output a graph of electrostatic force vs position between two particles. To start your program, make sure you import the following function libraries:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

For the first method, you should take your two mass values to be  $6 \cdot 10^{24}$  kg, and  $2 \cdot 10^{20}$  kg (hint: in python, the exponent symbol is `**`). Similarly to the last programming assignment, you should create a list of position values by using the following line of code:

```
position_values = np.linspace(50000, 100000, 50000)
```

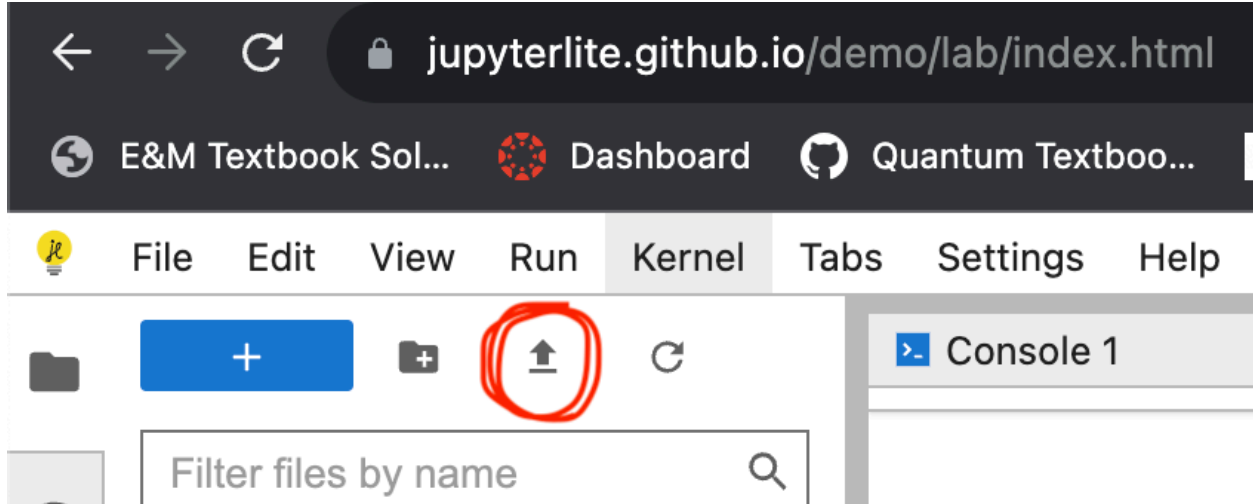
Using the equation for gravitational force you learned in mechanics, you should create a variable for your y-axis values. Using both your position values and your force equation, you should plot a graph in a similar manner as in the last assignment. Once you have run your code and correctly output a graph for gravitational force as a function of position, you can start working on part 2!

Part 2)

To start off, copy and paste your code from part 1 into a new cell, as you will be outputting both graphs together in this part. For your second method, you will be using the CSV file of data values I have provided to create your graph. In order to get data from your CSV file, you should follow these steps.

(1)

Add the csv file into your jupyter program using the upload file button.



Then, use the following line of code to read all the data from the csv file and access each column to be your x and y values:

```
data = pd.read_csv('electrostatic_data.csv')
```

```
x_values = data['position (cm)']
```

```
y_values = data['electrostatic force (N)']
```

(2)

Once you have completed the block of indented code in your method, you need to call your method in a new line (not indented!). At this point, when you run your code, it should output the same graph as in part 1, as well as a new graph for electrostatic force as a function of position!

Make sure both of your graphs have a title and axis labels for full credit.

### TAKE-HOME ASSIGNMENT

If you were not able to complete the activity by the end of class, you can continue working with your group outside of class to finish making your graphs. Remember, while you are encouraged to

work together, each of you should make your own submissions for credit. Once you have submitted your graph and spreadsheet file, you will complete an individual reflection below about your experience with the activity.

1. What was your level of knowledge of programming before this lesson?
2. Do you feel confident in your ability to write code and create graphs in python after the activity in class?
3. What part of the assignment did you / your group struggle with the most?

This homework will not be graded, so be as honest about your experience as possible! Use the space on the next page to write your reflection.



## Appendix C – Ideal Code

### *Lesson 3*

```
import matplotlib.pyplot as plt
import numpy as np
import math

# Given values
m = 2
amplitude = 0.2
k = 10

w = (k / m) ** 0.5

period = (2 * np.pi) / w

time_values = np.linspace(0, period, 100)

# Part 1: Creating energy equations

kinetic_energy = (1/2) * k * (amplitude**2) * ((np.sin(w * time_values))**2)
potential_energy = (1/2) * k * (amplitude**2) * ((np.cos(w * time_values))**2)
total_energy = kinetic_energy + potential_energy

# Part 2: Making graphs

plt.plot(time_values, kinetic_energy)
plt.plot(time_values, potential_energy)
plt.plot(time_values, total_energy)

plt.xlabel("Time")
plt.ylabel("Energy")

# Output graphs

plt.show()
```



## Lesson 4

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

def gforce():
    m1 = 6 * 10**24
    m2 = 2 * 10**20
    gravitational_constant = 6.67 * 10**(-11)

    position_values = np.linspace(50000,1000000,50000)
    gforce_values = gravitational_constant * m1 * m2 / (position_values**0.5)

    plt.plot(position_values,gforce_values)
    plt.xlabel("distance (m)")
    plt.ylabel("gravitational force")
    plt.show()

def electrostatic_force():

    data = pd.read_csv('electrostatic_data.csv')
    x_values = data['position (cm)']
    y_values = data['electrostatic force (N)']

    plt.plot(x_values,y_values)
    plt.xlabel("position (m)")
    plt.ylabel("electrostatic force (N)")
    plt.show()

electrostatic_force()
gforce()
```

## Appendix D - Rubrics

### *Lesson 1*

CRITERIA	4	3	2	1
Formatting spreadsheet	Column A contains all of the relevant variable names as described in the assignment. Column B contains all of the correct values.	Columns A and B have at most one error.	Columns A and B have two or more errors.	No attempt was made to fill in columns A and B.
Creating a formula for the height of the cliff	Your formula correctly outputs the height of the cliff by referencing all of the relevant cells.	Your formula for calculating the height of the cliff has minor errors that may or may not cause an incorrect response for your height value.	Your formula for calculating the height of the cliff has major errors that cause an incorrect response for your height value.	No attempt was made to create a formula for calculating the height of the cliff.
Creating a position-time chart	Column A correctly contains time values ranging from 0 to 5 seconds in 0.1 second intervals. Your formula for position as a function of time references the corresponding cell in column A and correctly calculates the position based on the given values.	Column A may be correct, but your formula for calculating the position values has minor errors that may or may not cause incorrect values.	While column A is filled with values for time, there are major errors in your spreadsheet that cause incorrect position values.	No attempt was made to fill column A with time values or column B with position values.
Creating a graph based on your position-time chart	Your graph has the correct title, labels, and axes. The graph correctly demonstrates the motion of the ball as it falls from a cliff of a particular height based on your time and position chart.	Your graph is present but your title, labels, or axes may be missing or incorrect. The graph is mostly accurate for demonstrating the motion of the ball.	Your graph is present but is missing two or more of the necessary features. The graph is inaccurate at demonstrating the motion of the ball.	No attempt was made to create a graph.

*Lesson 2*

CRITERIA	4	3	2	1
x and y-velocity values	Your formula for velocity of an object based on its changing position values correctly outputs values in the x and y directions. All of your velocity values align with the ideas talked about in class regarding projectile motion.	Your formula for velocity of an object based on its changing position values somewhat correctly outputs values in the x and y directions. Your formula may have minor errors that cause this.	Your formula for velocity of an object based on its changing position values incorrectly outputs values in the x and y directions. Your formula has major errors that cause this.	No attempt was made at developing a formula for velocity of an object based on its changing position values in the x and y directions.
x and y-acceleration values	Your formula for acceleration of an object based on its changing position values correctly outputs values in the x and y directions. All of your velocity values align with the ideas talked about in class regarding projectile motion.	Your formula for acceleration of an object based on its changing position values somewhat correctly outputs values in the x and y directions. Your formula may have minor errors that cause this.	Your formula for acceleration of an object based on its changing position values incorrectly outputs values in the x and y directions. Your formula has major errors that cause this.	No attempt was made at developing a formula for acceleration of an object based on its changing position values in the x and y directions.
Position-time graph	Your graph has the correct title, labels, and axes, correctly represents projectile motion, and shows creativity.	Your graph has minor errors but still somewhat accurately represents projectile motion. The graph shows some creativity.	Your graph has major errors, inaccurately represents projectile motion, and shows little to no creative effort.	No effort was made to create a graph.
Velocity-time graph	Your graph has the correct title, labels, and axes, correctly represents projectile motion, and shows creativity.	Your graph has minor errors but still somewhat accurately represents projectile motion. The graph shows some creativity.	Your graph has major errors, inaccurately represents projectile motion, and shows little to no creative effort.	No effort was made to create a graph.
Acceleration-time graph	Your graph has the correct title, labels, and axes, correctly represents projectile motion, and shows creativity.	Your graph has minor errors but still somewhat accurately represents projectile motion. The graph shows some creativity.	Your graph has major errors, inaccurately represents projectile motion, and shows little to no creative effort.	No effort was made to create a graph.

### Lesson 3

CRITERIA	4	3	2	1
Finding values for angular frequency and period	You correctly write equations for angular frequency and period, and assign them to variables that you create. These equations should reference the variables that contain the values given in the assignment.	Your equations for angular frequency and period are mostly correct, but they may have some minor errors.	Your equations for angular frequency and period are inaccurate, and contain major errors.	No attempt was made to write equations for angular frequency and period based on the given conditions.
Using built-in functions, write equations for kinetic, potential, and total energy	Using Python's built-in trigonometric functions ( <code>math.sin()</code> and <code>math.cos()</code> ), you correctly derive equations for kinetic, potential, and total energy. Your equations should reference both given variables as well as the variables you wrote equations for.	Your equations for kinetic, potential, and total energy are mostly correct, but they may have some minor errors.	Your equations for kinetic, potential, and total energy are inaccurate, and contain major errors.	No attempt was made to write equations for kinetic, potential, and total energy based on the given conditions.
Formatting your plot using Python's matplotlib capabilities	You should correctly utilize the sub-functions described in the assignment to create and format graphs for your three equations.	You utilized some of the sub-functions described in the assignment correctly, but some are incorrect or missing.	You did not use many of the sub-functions described in the assignment, which caused your graph formatting to be incorrect.	No attempt was made to use any sub-functions to format your graph.
Graphing your energy equations	Your graph accurately demonstrates the potential, kinetic, and total energy associated with spring-based simple harmonic motion.	Your graph is missing some elements, such as correct labels, axes, or a title, or incorrectly demonstrates the different energies of SHM in some way. The missing elements are minor, as the graph is still fairly accurate.	Your graph has major errors or many missing elements. As a result, the graph is an inaccurate representation of the energies associated with SHM.	No attempt was made to create a graph plotting the three energy equations.

*Lesson 4*

CRITERIA	4	3	2	1
Gravitational Force Method	You correctly imported the necessary libraries, created an equation for gravitational force, and formatted a plot for your data.	You had minor errors in your code that did not have any major impact on your graph. Your graph may be missing some necessary elements, or your equation may be somewhat incorrect.	You had major errors in your code that greatly impacted the accuracy of your graph.	No attempt was made at creating an equation and graph for gravitational force as a function of position.
Electrostatic Force Method	You correctly imported the necessary libraries, successfully uploaded the given CSV file, and stored the data from the file into variables that allowed you to create and format your final graph.	You had minor errors in your code that did not have any major impact on your graph. Your graph may be missing some necessary elements, or your equation may be somewhat incorrect.	You had major errors in your code that greatly impacted the accuracy of your graph.	No attempt was made at creating a graph for electrostatic force as a function of position.
Code Documentation	Your code is clear and well-structured, including numerous comments and relevant variable names.	Your code includes comments and relevant variables, but it may be a bit unclear or unstructured.	Your code has very little comments, or variable names are not very relevant. The code may be difficult to follow.	No attempt was made at following conventional documentation methods as described in the assignment.

## References

- AAPT Undergraduate Curriculum Task Force. (2016, October). AAPT Recommendations for the Undergraduate Physics Laboratory Curriculum
- Adkins, K., & Birriel, J. (2021, December 21). Introducing computation: A laboratory approach. PICUP. <https://www.compadre.org/PICUP/resources/Entry.cfm?ID=124835>
- American Physical Society. (2023). Effective practices for physics programs (EP3). [ep3guide.org](https://ep3guide.org). <https://ep3guide.org/>
- Birriel, J., & Birriel, I. (2021, December 17). Spreadsheet computation in Introductory Physics Labs. PICUP. <https://www.compadre.org/PICUP/resources/Entry.cfm?ID=120053>
- Bowen, R. S. (2017). Understanding by Design. Vanderbilt University Center for Teaching. <https://cft.vanderbilt.edu/understanding-by-design/>.
- Caballero, M., Kohlmyer, M. Schatz, M. (2012, August 14). Implementing and assessing computational modeling in introductory mechanics. *Physical Review Physics Education Review*. <https://journals.aps.org/prper/abstract/10.1103/PhysRevSTPER.8.020106>

Computational Physics in the Undergraduate Physics Curriculum. *American Association of Physics Teachers*.

[https://www.aapt.org/resources/upload/aapt\\_uctf\\_compphysreport\\_final\\_b.pdf](https://www.aapt.org/resources/upload/aapt_uctf_compphysreport_final_b.pdf)

Cummings, E. (2023). The 10 Most Common Accommodations Found on a Special Education IEP. Behaviorist. <https://www.behaviorist.com/what-are-accommodations-in-special-education/>

Eva, H. C. (2014). Group work as an incentive for learning – students' experiences of group work. *Frontiers in Psychology*, 5.

Fink, L. D. (2005). Integrated Course Design. Manhattan, Kansas; IDEA.

Glennon, M., Weissman, H., Twarek, B., Dunton, S., and Childs, J.(2023). State of Computer Science Education. *Computer Science Teacher's Association*.

[https://advocacy.code.org/2023\\_state\\_of\\_cs.pdf](https://advocacy.code.org/2023_state_of_cs.pdf)

Gregory, J., & Rehak, A. (2017, February). Creating and using rubrics. *Center for Teaching and Learning*. <https://ctl.iupui.edu/Resources/Assessing-Student-Learning/Creating-and-Using-Rubrics>

Guzdial, M. (2012, October 22). Why Isn't There More Computer Science in U.S. High Schools?. *Communications of the ACM*. <https://cacm.acm.org/blogs/blog-cacm/156531-why-isnt-there-more-computer-science-in-us-high-schools/fulltext>

High school introductory physics instructional guidelines. Department of Elementary and Secondary Education. (2022, April). <https://www.doe.mass.edu/stem/ste/hs-intro-phys.pdf>

Home Page: Next generation science standards. Home Page | Next Generation Science Standards. (n.d.). <https://www.nextgenscience.org/>

- Law, B. & Eckes, M. (1995). *Assessment and ESL*. Manitoba, Canada: Peguis Publishers.
- Referenced in <https://files.eric.ed.gov/fulltext/EJ1101956.pdf> , follow link for particular statement on page 15
- Massachusetts Department of Elementary and Secondary Education. (2014). *Massachusetts' adaptation of Next Generation Science Standards*. Massachusetts' Adaptation of Next Generation Science Standards - Science, Technology/Engineering, and Mathematics (STEM). <https://www.doe.mass.edu/stem/standards/NGSS-MAAComparison.html#:~:text=The%20MA%20Draft%20Revised%20STE,ideas%20and%20practices%20in%20standards>
- McNeil, L., Heron, P. (2017, November 1). Preparing physics students for 21st-century careers. *Physics Today*, 70(11), 39-43.
- Nguyen, T. (2013). *Understanding Different Learning Theories for Teaching in the 21st Century*. Waynesville, North Carolina: Association for the Advancement of Computing in Education.
- Northern Illinois University Center for Innovative Teaching and Learning. (2012). *Formative and summative assessment. Instructional guide for university faculty and teaching assistants*. <https://www.niu.edu/citl/resources/guides/instructional-guide>
- Sheldrake, R. (2016). Confidence as motivational expressions of interest, utility, and other influences: Exploring under-confidence and over-confidence in science students at secondary school. *International Journal of Educational Research*, 76, 50-65.
- Simonson M., Smaldino, S, Albright, M. and Zvacek, S. (2000). *Assessment for distance education* (ch 11). *Teaching and Learning at a Distance: Foundations of Distance Education*. Upper Saddle River, NJ: Prentice-Hall. Referenced in



<https://files.eric.ed.gov/fulltext/EJ1101956.pdf> , follow link for particular statement on page 15

Tanner K. D. (2012). Promoting student metacognition. *CBE life sciences education*, 11(2), 113–120. <https://doi.org/10.1187/cbe.12-03-0033>

Winking, D. (1997). Critical issue: Ensuring equity with alternative assessments [online document]. NCREL (North Central Regional Educational Laboratory), Oak Brook: IL. Available online: <http://www.ncrel.org/sdrs/areas/methods/assment/as800.htm>

Ying, Z., Zeqing Z., Chenchen, P., Lulu, Z. (2021). Self-Esteem and Academic Engagement Among Adolescents: A Moderated Mediation Model. *Frontiers in Psychology*, 12.

Young, Thomas A. Engaging Students In Energy Related Topics Using Contextualized Lessons In Physics. : Worcester Polytechnic Institute, 2020.

Zimmerman, T. (2020, September 21). So you've decided to use python - what's next? PICUP. <https://www.compadre.org/PICUP/resources/Entry.cfm?ID=119748>