Coding Integration into High School Physics using STEMcoding Project Resources
2.5 week intermediate-level course (2.0 grad credits)

Instructor: Prof. Chris Orban
Contact Info: 614-557-9387, e-mail: orban@physics.osu.edu

Course Meetings:
Zoom videoconferences (typically at 1pm EST unless otherwise noted)

Dates: June 29 - July 15, 2020

Final exam: No final exam
Course web page: `http://stemcoding.osu.edu`

Intermediate level?

This course occurs after an absolute-beginner-level two-week 1.0 credit hour course taught by Prof. Orban June 15 - June 26, 2020. The activities in the absolute-beginner course are simple enough that they could be used (and have been used) in high school physical science classes. The activities in this "intermediate level" course are designed for non-AP physics level courses, and college algebra-based introductory physics courses.

Compared to the absolute beginner course, this course counts for 2.0 credit hours which means that the pace of the course is twice as fast as the absolute beginner course and we will cover about twice as many STEMcoding activities.

Participating in the absolute beginner level course is **not** a requirement for taking this intermediate level course. Experienced physics teachers with little to no programming experience are welcome to skip the absolute beginner level course and start with this intermediate level course, so long as they have plenty of time during June 29 - July 15 to participate. You will find that your physics knowledge will often be helpful in modifying the codes and noticing problems with the programs.

Teachers who have recently been assigned to teach physics for the first time may struggle with the pace of this course and find the absolute beginner level course to be much more appropriate and comfortable.

Specific Goals/Rationale

The goal of this particular course is to show teachers how computer programming exercises can be integrated into high school physics (AP or non-AP) in a simple, fun and engaging way. For a variety of reasons, this course is not intended to take the place of "modeling instruction" workshops by the American Modeling Teachers Association (AMTA) in any way.

The basis of this course is a set of computer programming exercises that Prof. Orban developed (at least initially) for freshman physics courses on OSU's Marion campus. Over a thousand college and high school students, many of whom have never written a computer program before, have successfully completed these exercises. As discussed in Orban et al. 2018 (`https://aapt.scitation.org/doi/10.1119/1.5058449`) surveys of students at OSU Marion seem to indicate that the content is at an appropriate difficulty level, even for absolute beginner programmers.

Videoconferencing

We will regularly use Zoom videoconferencing as part of the course. Please download and install zoom on your laptop or computer by going to `http://zoom.us`. If you cannot get zoom working on a computer or laptop, you can use zoom on a smartphone or tablet instead. **You should be able to use the free version of zoom to participate in this course!**

Strings attached?

There is no specific requirement that you will use these activities in your courses in the coming academic year. Ultimately it is up to you to decide how pervasive you want to make computer programming activities in your own course. There are no "strings attached" to your participation in this course. There is no funded grant that requires you to you participate in an ongoing study.

Relevancy to Science, Computer Science and Math Standards

The Next Generation Science Standards (NGSS), which is the basis of most state science standards in the US, includes "computational thinking" as one of its core practices. For an in-depth discussion of how activities like these align with computational thinking please consult this paper: Orban & Teeling-Smith `https://arxiv.org/abs/1907.08079` This paper has been accepted to publication to *The Physics Teacher* and may have already been printed there.

Regarding the specific statements in the NGSS that outline important science learning objectives, the activities that will be discussed in this course do align with many of these statements. A google document that summarizes the alignment between NGSS and the STEMcoding activities in this course is accessible from this link: `http://go.osu.edu/ngss`

Increasingly, states are adopting or creating K12 computer science (CS) standards. Typically these standards are derived from a national set of standards approved by CSTA. For the most part, these state CS standards can be interpreted as a blanket permission to include computer science and computational thinking oriented activities in K12 science and math classes. We believe our activities line up well against state and national CS standards to the point where one could use these activities as the basis for a computer science course. We are happy to discuss more specifics of CS standards if you are interested.

Regarding math standards, the activities in this course involve algebra 1, algebra 2 and trigonometry level concepts including working with 2D plots with x and y axes. The STEMcoding activities in this course do NOT involve or require any calculus.

Expected Learning Outcomes

1. Teachers will learn how to use a computer code to solve physics problems in an iterative way
2. Teachers will learn about potential pitfalls and common mistakes that students make in completing physics-focused computer programming exercises
3. Teachers will learn some of the limitations of using a computer code to iteratively solve physics problems (e.g. energy conservation, lack of perfect agreement with analytic expectations)
4. Teachers will become familiar with assessment questions designed to probe student conceptual knowledge gains from completing physics-focused computer programming exercises
5. Teachers will become familiar with the `http://stemcoding.osu.edu` learning management system (which is free to use and will remain free)

How the Learning Objectives will be met

1. Teachers will complete a set of physics-focused computer programming exercises that solve physics problems in an interative way, and often in an interactive way
2. After completing each exercise, teachers will read through a set of detailed instructor notes outlining potential pitfalls and common mistakes
3. Many of the exercises include comparisons to analytic expectations. Teachers will learn the limitations of the codes by completing the exercises and in discussions with Prof. Orban
4. Teachers will complete pre and post assessments for each exercise. These assessments are the same that students would complete.
5. After completing the exercises teachers will learn to set up a course in the `http://stemcoding.osu.edu` learning management system

Texts and Required Material

There is no required text for this course. All of the exercises are available at `http://stemcoding.osu.edu`

STEMcoding Learning Management System (stemcoding.osu.edu)

OSU created a custom-built learning management system for Prof. Orban's coding activities which is available at `http://stemcoding.osu.edu`. Please register an e-mail address on `http://stemcoding.osu.edu` and use the course join key AAPTSummer2020Intermediate You can register any e-mail you want with the system. It does not have to be your school e-mail. At the end of the course you will be given "teacher" status on the learning management system so you can set up your own courses and grade student submissions. This service is provided free-of-charge.

This learning managment system has been used successfully on various operating systems, including chromebooks and iPads.

What programming language is involved?

The activities in this course involve modifying and running javascript programs. For people with experience working with C, C++ or Java code, this language will seem very familiar. We do not currently use python or vpython in our activities. Python uses indentation instead of curly brackets to define the logic of the program, which means that just adding a space to the code in the wrong place could change the behavior of the program and lead to confusion. (However, many people argue that the role of indentation in python helps students avoid writing sloppy looking code.) We may set up a python version of our activities in the future if there is interest.

Participants with Disabilities

Any teacher who feels that he or she may need an accommodation based on the impact of a disability should contact me to discuss their specific needs.

Grading:

Teachers who complete required exercises will earn an A in the course. The exercises are available on `http://stemcoding.osu.edu`

Required exercises:

1. Planetoids
2. Lunar Descent
3. Bellicose Birds
4. Planetoids with momentum!
5. Pong
6. Bonk.io (elastic collisions + gravity)
7. Planetoids with torque!
8. Planetoids with a spring!
9. Particle accelerator

Optional: STEMcoding Slack Channel

In the previous summers that this course has been offered, teachers have often wanted to talk to each other about the exercises during the course when they have run into trouble or to check with others to see if they are seeing the same technical issue. To facilitate this Prof. Orban has created a “slack channel” at `http://stemcoding.slack.com` to allow you to talk to each other. Prof. Orban would need to manually add you to that channel to get access. It can be with whatever e-mail you prefer.

Tentative Course Schedule

(all times are Eastern Standard Time)

	Dates	Topic / Item	Assignments due
Week 1	June 29	video chat at 1pm	Hello world / troubleshooting
	June 30	video chat at 1pm	Planetoids
	July 1	video chat at 1pm	Lunar Descent
	July 2	video chat at 1pm	Bellicose birds
	July 3	Optional video chat at 1pm	
Week 2	July 6	video chat at 1pm	Planetoids with momentum
	July 7	video chat at 1pm	Pong
	July 8	video chat at 1pm	Bonk.io
	July 9	video chat at 1pm	Planetoids with torque
	July 10	video chat at 1pm	Planetoids with a spring
Week 3	July 13	video chat at 1pm	Particle accelerator
	July 14	video chat at 1pm	
	July 15	no video chat	Final programs due