CodeHS

Introduction to Computer Science in Python
Common Core State Standards for Math
Standards Alignment Overview

The CodeHS Introduction to Computer Science in Python curriculum teaches the foundations of computer science and basic programming, with an emphasis on helping students develop logical thinking and problem solving skills. This document is an overview of how the Introduction to Computer Science in Python course aligns with the Common Core State Standards for Mathematics.

Standards for Mathematical Practice

1. **Make sense of problems and persevere in solving them.**
   Troubleshooting computing systems is a core concept of the Introductory Python course. Computing systems might not work as expected because of problems in the software. Students are expected to identify problems in their programs and fix them.

   Errors are opportunities to learn from mistakes. In the CodeHS Intro Python course, students continually test programs that that they build using Tracy the Turtle and Python, and these programs will invariably have bugs. Students are encouraged during testing to start by explaining to themselves the context the problem they are trying to solve and looking for entry points to its solution. They plan a solution pathway rather than jumping into a solution attempt with the issues they are experiencing. They persevere in solving them by checking their Python by using the step debugger and other debugging strategies.

2. **Reason abstractly and quantitatively.**
   Nearly every computer science course deals with various levels of abstraction, and the Introduction to Computer Science in Python is no exception. Students create their own Python functions using top down design with meaningful function names to abstract away various algorithms in their programs. The Intro Python course encourages students to make sense of quantities and their relationships in problem situations. Students need to consider design layouts with many quantitative decisions in the graphics, animation, and game exercises.

3. **Construct viable arguments and critique the reasoning of others.**
   Intro Python students are able to analyze situations by breaking them into cases so

they can create programs that solve problems or express their creativity. They justify their conclusions and decisions for choices that they make in their programs, communicate them to others, and respond to the arguments of others about advantages and disadvantages for developing their programs in a certain way.

Computing has had significant impacts in several fields. In this course, students learn first hand the impacts computing has on digital drawings, gaming, and interpreting large text passages. Students also learn the ethical considerations of sharing their code with others, and searching online for solutions to CodeHS exercises.

4. **Model with mathematics.**
   Students in the CodeHS Intro Python course use geometry to solve graphics, animation and game design layout problems. Students in the course can also apply what they know and are comfortable making assumptions and approximations to simplify a complicated designs, realizing that these may need revision later. Finally, students solve many classical mathematics problems (e.g., making a calculator, unit conversion, creating sets) with their programs.

5. **Use appropriate tools strategically.**
   Students might use pencil and paper, concrete models, a ruler, a calculator, and other tools to create their programs in the Intro Python course. Students detect possible errors in their code by strategically using estimation and other mathematical knowledge.

6. **Attend to precision.**
   Storing, transforming, and visualizing data are all taught as part of the Introduction to Python course. Students learn to use various types of data structures to store data, as well as how to select the proper data structure to model a problem. Students have the option to write programs that manipulate data, find elements in data structures, remove elements from data structures, and even guess who wrote various texts in the Supplemental Units of the course.

7. **Look for and make use of structure.**
   General algorithms in programs create structures that can be used repeatedly. Algorithms and programming are the central focus of the Intro Python course. Students learn the core principles of developing their own algorithms and implementing them in the Python programming language. Algorithms, variables, control, modularity, and program development are all taught in this course.

8. **Look for and express regularity in repeated reasoning.**
   In the Intro Python course, students notice if calculations are repeated, and look both for general methods and shortcuts to make make their code more concise and reusable.

Standards for Mathematical Content

Introduction to Computer Science in Python
 The CodeHS introduction to Python course teaches the fundamentals of computer programming as well as some advanced features of the Python language. Students use what they learn in this course to build simple console-based games. This course is equivalent to a semester-long introductory Python course at the college level.
Standards for Mathematical Content Addressed
6.RP.1 Understand the concept of a ratio and use ratio language to describe a ratio relationship between two quantities. **6.RP.3** Use ratio and rate reasoning to solve real-world and mathematical problems, e.g., by reasoning about tables of equivalent ratios, tape diagrams, double number line diagrams, or equations. **6.NS.3** Fluently add, subtract, multiply, and divide multi-digit decimals using the standard algorithm for each operation. **6.NS.5** Understand that positive and negative numbers are used together to describe quantities having opposite directions or values (e.g., temperature above/below zero, elevation above/below sea level, credits/debits, positive/negative electric charge); use positive and negative numbers to represent quantities in real-world contexts, explaining the meaning of 0 in each situation. **6.NS.6c** Find and position integers and other rational numbers on a horizontal or vertical number line diagram; find and position pairs of integers and other rational numbers on a coordinate plane. **6.NS.7** Understand ordering and absolute value of rational numbers. **6.NS.8** Solve real-world and mathematical problems by graphing points in all four quadrants of the coordinate plane. Include use of coordinates and absolute value to find distances between points with the same first coordinate or the same second coordinate. **6.EE.2** Write, read, and evaluate expressions in which letters stand for numbers. **6.EE.2c** Evaluate expressions at specific values of their variables. Include expressions that arise from formulas used in real-world problems. Perform arithmetic operations, including those involving whole- number exponents, in the conventional order when there are no parentheses to specify a particular order (Order of Operations). **6.EE.6** Use variables to represent numbers and write expressions when solving a real-world or mathematical problem; understand that a variable can represent an unknown number, or, depending on the purpose at hand, any number in a specified set. **6.EE.8** Write an inequality of the form $x > c$ or $x < c$ to represent a constraint or condition in a real-world or mathematical problem. Recognize that inequalities of the form $x > c$ or $x < c$ have infinitely many solutions; represent solutions of such inequalities on number line diagrams. **6.G.3** Draw polygons in the coordinate plane given coordinates for the vertices; use coordinates to find the length of a side joining points with the same first coordinate or the same second coordinate. Apply these techniques in the context of solving real-world and

mathematical problems.
7RP.2 Recognize and represent proportional relationships between quantities.
7.NS.1 Apply and extend previous understandings of addition and subtraction to add and subtract rational numbers; represent addition and subtraction on a horizontal or vertical number line diagram.
7.NS.1b Understand p + q as the number located a distance |q| from p, in the positive or negative direction depending on whether q is positive or negative. Show that a number and its opposite have a sum of 0 (are additive inverses). Interpret sums of rational numbers by describing real-world contexts.
7.NS.3 Solve real-world and mathematical problems involving the four operations with rational numbers.
7.EE.3 Solve multi-step real-life and mathematical problems posed with positive and negative rational numbers in any form (whole numbers, fractions, and decimals), using tools strategically. Apply properties of operations to calculate with numbers in any form; convert between forms as appropriate; and assess the reasonableness of answers using mental computation and estimation strategies.
7.EE.4 Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.
7.SP.5 Understand that the probability of a chance event is a number between 0 and 1 that expresses the likelihood of the event occurring. Larger numbers indicate greater likelihood. A probability near 0 indicates an unlikely event, a probability around 1/2 indicates an event that is neither unlikely nor likely, and a probability near 1 indicates a likely event.
8.EE.1 Know and apply the properties of integer exponents to generate equivalent numerical expressions.
8.F.1 Understand that a function is a rule that assigns to each input exactly one output. The graph of a function is the set of ordered pairs consisting of an input and the corresponding output.
N-Q.1 Use units as a way to understand problems and to guide the solution of multi-step problems; choose and interpret units consistently in formulas; choose and interpret the scale and the origin in graphs and data displays.
A-SSE.1 Interpret expressions that represent a quantity in terms of its context.
F-IF.1 Understand that a function from one set (called the domain) to another set (called the range) assigns to each element of the domain exactly one element of the range. If f is a function and x is an element of its domain, then f(x) denotes the output of f corresponding to the input x. The graph of f is the graph of the equation y = f(x).
F-BF.1 Write a function that describes a relationship between two quantities.
F-BF.1a Determine an explicit expression, a recursive process, or steps for calculation from a context.
F.LE.5 Interpret the parameters in a linear or exponential function in terms of a context.