Computer Programming Honors

10^{th} - 12^{th}

Prepared by:

Danielle Vandenberghe

Superintendent of Schools:

Marie C. Cirasella, Ed.D.

Approved by the Midland Park Board of Education on ,

August 23, 2022

Born on Date August 22, 2022

Computer Programming Honors

Course Description: Computer Programming I Honors is a beginning programming class using Java and Python programming languages. The first semester will focus on learning the basics of programming through the Java programming language. The second semester will focus on learning the basics of programming through Python. Lastly students will work on a final “Passion” Project. This project will have students researching a problem and creating a technology solution. This class assumes that you have little or no programming experience. It provides a solid background in good programming techniques and introduces terminology using clear, familiar language. Throughout the class, computational thinking and problem-solving skills will also be looked at thoroughly as this is a key component in computer programming. Upon the completion of the class students will understand concepts used in programming and be able to modify and create simple Java and Python programs. This class serves as a great resource for future programming classes that students may take.

Course Sequence:

Unit 1: Java Programming – 18 weeks
Unit 2: Python Programming – 15 weeks
Unit 3 – Final Project – 3 weeks

Prerequisite: Intro to Computers or Approval from Computer Teacher/Principal

Midland Park Public Schools

Unit 1- Overview

[redacted]

Core Ideas: In this unit students will learn the basics of programming in Java. Students will learn about variables, conditionals, loops, arrays, classes and object-ordinated programming all in the Java programming language

Unit 1 - Standards

[redacted]

8.1.12.AP.3 Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to

address a societal issue.
8.1.12.AP.5 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
8.1.12.AP.6 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.

[redacted] 8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.

[redacted] 9.2.12.CAP.2 Develop college and career readiness skills by participating in opportunities such as structured learning experiences, apprenticeships, and dual enrollment programs.
9.2.12.CAP.8 Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, and drug tests) used by employers in various industry sectors.

[redacted]
9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8)
9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
[redacted] 9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

[redacted]

LGBTQ and Disabilities NJSA 18A:35- 4.35

Explore computer scientist in the LGBTQ community, including but not limited to Sofia Kovalevskaya, Alan Turing, Christopher Strachey, Peter Landin, Edith Windsor, Lynn Conway, Jon Hall, Sphie Wilson, Mary Ann Horton, and Audrey Tang

Amistad Law NJSA 18A:35- 4.43	Explore African-American computer scientist, including but not limited to Clarence Ellis, Melba Roy Mouton, Katherine Johnson, Mary Jackson, and Annie Easley

Holocaust Law NJSA 18A:35- 28
AAPI Law
NJSA 18A:25- 4.44
Midland Park Public Schools

Explore Jewish computer scientist, including but not limited to Joseph Weizenbaum, Larry Page, Hilary Putnam, Jon von Neumann, Sergery Brin, Saul Amarel and Norbert Wiener

Explore Asian-American/Pacific Islander computer scientist, including but not limited to Peter Tsaai, Nainoa Thompson, Flossie Wong-Staal, Ajay Bhatt, Min Chueh Chang, Roseli Ocamp-Friedmann, Steven Shih Chen, and Ching Wan Tang

[redacted]

Science
HS-ETS1-4 History
6.1.12.SE.14.a

Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within

and between systems relevant to the problem. Explore the various ways women, racial and ethnic minorities, the LGBTQ community, and individuals with disabilities have contributed to the American economy, politics and society

[redacted]

NJSLSA.SL1 Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.

NJSLSA.SL2 Integrate and evaluate information presented in diverse media and formats, including visually, quantitatively, and orally.

[redacted]

Self-Awareness
- Examining prejudices and biases
- Having a growth mindset

Self Management
- Exhibiting self-discipline and self-motivation
- Using planning and organizational skills

Social Awareness
- Taking others' perspectives
- Understanding the influences of organizations/systems on behavior

Relationship Skills
- Communicating effectively
- Practicing teamwork and collaborative problem-solving
- Showing leadership in groups

Responsible Decision Making
- Demonstrating curiosity and open-mindedness
- Recognizing how critical thinking skills are useful both inside & outside of school

Unit Essential Question(s):	Unit Enduring Understandings:
• How can we use programs to solve problems • In what ways are numbers used in the programs and apps you use most often • How are mathematical concepts being used in the programs and apps that you use most often? • How are appropriate variables chosen to represent a remote control? • How do the games we play simulate randomness? • Why is selection a necessary part of programming languages? • How does iteration improve programs and reduce the amount of program code necessary to complete a task? • What situations would warrant the use of one type of loop over another? • What responsibility do programmers have for the consequences of programs they create, whether international or not? • How can knowing standard algorithms be useful solving new problems?	• Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence. • To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values. • The way variables and operators are sequenced and combined in an expression determines the computed result. • Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values. • Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept. • When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

- While programs are typically designed to achieve a specific purpose, they may have unintended consequences.
- To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

Evidence of Learning

Formative Assessments:
- Do Now
- Teacher observations
- Questioning
- Quizzes
- Practice Programs
- Entry tickets
- Exit tickets
- Online games
- Discussions
- Homework

Summative/Benchmark Assessment(s):
- Projects
- Tests
- Chapter Review / Quizzes
- Chapter Assignments
- Labs
- Final Project

Alternative Assessments:
- Portfolio
- Projects
- Online tests / assignments

Resources/Materials: ▪ JGrasp ▪ Finch Robots - https://www.birdbraintechnologies.com/finch-start teaching/ ▪ Text Book: Introduction to Java Programming by Y. Daniel Liang ▪ http://wps.pearsoned.com/ecs_liang_ijp_10/244/62489/ 15997433.cw/index.html ▪ http://csunplugged.com/ ▪ www.codehs.com	Key Vocabulary: Algorithm, Assignment operator, Assignment statement, Casting, Constant, Data type, Declare variables, Decrement operator (--), Double type, Expression, Final keyword, Identifier, Increment operator (++), int type, Literal, Operands, Operator. Overflow, Postdecrement, Postincrement, Predecrement, Preincrement, Primitive data type, Pseudocode, Requirements specification, Scope of a variable, Specific import, System analysis, System design, Underflow, Variable, Wildcard import, Class, Object, Attributes, Instance variables, Methods, Constructor, New, Constructor signature or header, Method signature or header, Wrapper classes, Null, Math method: int abs(int), Math method: double abs(double), Math method: double pow(double, double), Math method: double sqt(double), Math.random(), Parameter list, Overloading, Call by value, NullPointerException, Non-static or object method, Static method, Return value, Strings, Index, Immutable, Substring, String method: int length(), String substring(int from, int to), String substring(int from), int indexOf(String str), boolean equals(String other), int compareTo(String other), String substring(index, index + 1), Autoboxing, Unboxing, (int) (Math.random()*range) + min, Block of Statements, Boolean expression, Complex Conditional,

and, Logical or, negation, Short circuit evaluation, Body of a Loop, for, For Each Loop, For Loop, Infinite Loop, Loop, Nested Loop, Out of Bounds error, Trace Code, while, While Loop, Class, Constructor, Public, Private, Object, Accessor, Mutator, Return, Void, Static, Method, Argument , Parameter, Method overloading, Method signature, Behavior, Instance, Instance variable, No-arg constructor, Null value, this Keyword, Data abstraction, Array, Element, Parallel arrays, Traversing, Array initializer, Index, Indexed variable, Array Length, For each

Lesson Name/Topic
Midland Park Public Schools

Suggested Pacing Guide

Student Learning Objective(s) Suggested Tasks/Activities: Day(s) to Complete

Conditional, DeMorgan's Law, else, else if , if , Logical

Primitive Types
▪ Call System class methods to generate output to the console
▪ Create string literals
▪ Identify the most appropriate data type category for a particular specification
▪ Declare variables of the correct types to

represent primitive data ☛ Evaluate arithmetic expressions in a program code. ☛ Evaluate what is stored in a variable as a result of an expression with an assignment statement ☛ Evaluate what is	stored in a variable as a result of an expression with an assignment statement. ☛ Evaluate arithmetic expressions that use casting. ☛ Students will learn the following topics: o Why Programming?	Why Java? o Variables and Data Types o Expressions and Assignment Statements o Compound Assignment Operators o Casting and Ranges of Variables ☛ Each lesson will	include vocabulary, notes and practice ☛ At the end of the unit students will work on practice programs, multiple choice questions, finch robot programs and a unit project. 10 days

Using Objects	☛ Explain the relationship between a class and an object. ☛ Identify, using its signature, the correct constructor being called. ☛ Create objects by calling constructions with and without parameters ☛ Define variables of the correct types to represent reference data. ☛ Call non-static void methods with and without parameters ☛ Call non-static non-void methods with or without parameters. ☛ Create String objects ☛ Call String methods ☛ Create Integer objects ☛ Call Integer methods ☛ Create Double objects ☛ Call Double methods ☛ Call static methods	☛ Students will learn the following topics: o Objects: Instances of Classes o Creating and Storing Objects (Instantiation) o Calling a Void Method o Calling a Void Method with Parameters o Calling a Non-void Method o String Objects: Concatenation, Literals and More o String Methods o Wrapper Classes: Integer and Double o Using the Math Class ☛ Each lesson will include vocabulary, notes and practice ☛ At the end of the unit students will work on practice programs,	15 days

Boolean Expressions and if Statements ☛ Evaluate expressions that use the Math class methods ☛ Evaluate Boolean expressions that use relational operators in	program code ☛ Represent branching logical processes by using conditional statements. ☛ Compare and contrast equivalent Boolean expressions ☛ Compare object references using Boolean expressions in program code. multiple choice questions, finch robot	programs and a unit project. ☛ Students will learn the following topics: o Boolean Expressions o if Statements and Control Flow o if-else Statements o else if Statements o Compound Boolean Expressions o Equivalent Boolean Expressions o Comparing Objects	☛ Each lesson will include vocabulary, notes and practice ☛ At the end of the unit students will work on practice programs, multiple choice questions, finch robot programs and a unit project. 15 days

processes using a while loop
- Identify standard algorithms
- Modify standard algorithms
- Develop an algorithm
- Represent iterative processes using a for loop
- Represent nested iterative processes
- Compute statement execution counts and informal run-time comparison of iterative statements.
- Students will learn the following topics:

o while Loops
o for Loops
o Developing Algorithms Using Strings
o Nested Iteration
o Informal Code Analysis
- Each lesson will include vocabulary, notes and practice
- At the end of the unit students will work on practice programs, multiple choice questions, finch robot programs and a unit project.

20 days

Iteration ■ Represent iterative

Writing Classes	■ Designate access and visibility constraints to classes, data, constructors, and methods. ■ Designate private visibility of instance variables to encapsulate that attributes of an object. ■ Define instance variables for the attributes to be initialized through the constructors of a class. ■ Describe the functionality and use of program code through comments. ■ Define behaviors of an object through non-void methods without parameters written in a class. ■ Define behaviors of an object through void methods with or without parameters written in a class.	■ Students will learn the following topics: o Anatomy of a Class o Constructors o Documentation with Comments o Accessor Methods o Mutator Methods o Writing Methods o Static Variables and Methods o Scope and Access o this Keyword o Ethical and Social Implications of Computing Systems ■ Each lesson will include vocabulary, notes and practice	15 Days

■ Define behaviors of an object through non-void methods with parameters written in a class.
■ Define behaviors of a class through static methods.

■ Explain where variables can be used in the program code.
■ Evaluate object reference expressions that use the keyword this.
■ Explain the ethical and social implications of computing systems.

Array ■ Represent collections of related primitive or object reference data using one-dimensional array objects.
■ Traverse the elements in a 1D array
■ Traverse the elements in a 1D array

object using an enhanced for loop.
- Identify standard algorithms
- Modify standard algorithms
- Develop an algorithm.
- At the end of the unit students will work on practice programs, multiple choice questions, finch robot programs and a unit project.

o Traversing Arrays
o Enhanced for Loop for

15 Days

- Students will learn the following topics:

o Array Creation and Access

Arrays
o Developing Algorithms Using Arrays
- Each lesson will include vocabulary, notes and practice
- At the end of the unit students will work on practice programs, multiple choice questions, finch robot programs and a unit project.

Teacher Notes: Unit tests will be given for sections 1-3 and 4-5. All sections will be on the midterm or Arrays will be tested by itself.

Additional Resources:

Differentiation/Modification Strategies

Students with Disabilities	Learners	Students	Students
English Language	Gifted and Talented	Students at Risk 504	

• Allow errors • Rephrase questions, directions, and explanations • Allow extended time to answer questions and permit drawing as an explanation • Accept participation on any level, even one word • Consult with Case Managers and follow IEP accommodations/ modifications	• Assign a buddy, same language or English speaking • Allow errors in speaking • Rephrase questions, directions, and explanations • Allow extended time to answer questions • Accept participation at any level, even one word	• Provide extension activities • Build on students' intrinsic motivation • Consult with parents to accommodate students' interests in completing tasks at their level of engagement	• Provide extended time to complete tasks • Consult with Guidance Counselors and follow I&RS procedures/action plans • Consult with other members of the 7th grade team for specific behavior interventions • Provide rewards as necessary	• Allow errors • Rephrase questions, directions, and explanations • Allow extended time to answer questions and permit drawing as an explanation • Accept participation on any level, even one word • Consult with Case Managers and follow IEP accommodations/ modifications • Assign a buddy, same language or English speaking

Midland Park Public Schools

Unit 2 - Overview

[redacted]

Core Ideas: This course teaches the fundamentals of computer programming as well as some advanced features of the Python 3 language. By the end of this course, students build a simple console-based game and learn material equivalent to a semester college introductory Python course

Unit 2- Standards

[redacted]

8.1.12.AP.3 Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.

8.1.12.AP.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

8.1.12.AP.5 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

8.1.12.AP.6 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback

from users.

[redacted] 8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.

[redacted]

9.2.12.CAP.2 Develop college and career readiness skills by participating in opportunities such as structured learning experiences, apprenticeships, and dual enrollment programs.

9.2.12.CAP.8 Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, and drug tests) used by employers in various industry sectors.

[redacted]

9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).

9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).

9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8)

9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).

[redacted] 9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

[redacted]

LGBTQ and Disabilities NJSA 18A:35- 4.35
Explore computer scientist in the LGBTQ community, including but not limited to Sofia Kovalevskaya, Alan Turing, Christopher Strachey, Peter Landin, Edith Windsor, Lynn Conway, Jon Hall, Sphie Wilson, Mary Ann Horton, and Audrey Tang

Amistad Law NJSA 18A:35- 4.43	Explore African-American computer scientist, including but not limited to Clarence Ellis, Melba Roy Mouton, Katherine Johnson, Mary Jackson, and Annie Easley

Holocaust Law NJSA 18A:35- 28
AAPI Law
NJSA 18A:25- 4.44
Midland Park Public Schools

Explore Jewish computer scientist, including but not limited to Joseph Weizenbaum, Larry Page, Hilary Putnam, Jon von Neumann, Sergery Brin, Saul Amarel and Norbert Wiener

Explore Asian-American/Pacific Islander computer scientist, including but not limited to Peter Tsaai, Nainoa Thompson, Flossie Wong-Staal, Ajay Bhatt, Min Chueh Chang, Roseli Ocamp-Friedmann, Steven Shih Chen, and Ching Wan Tang

[redacted]

Science
HS-ETS1-4 History
6.1.12.SE.14.a
Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem. Explore the various ways women, racial and ethnic minorities, the LGBTQ community, and individuals with disabilities have contributed to the American economy, politics and society

[redacted]

NJSLSA.SL1 Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.

NJSLSA.SL2 Integrate and evaluate information presented in diverse media and formats, including visually, quantitatively,

and orally.

[redacted]

Self-Awareness
- Examining prejudices and biases
- Having a growth mindset

Self Management
- Exhibiting self-discipline and self-motivation
- Using planning and organizational skills
- Taking others' perspectives
- Understanding the influences of organizations/systems on behavior

Social Awareness
- Communicating effectively
- Practicing teamwork and collaborative problem-solving
- Showing leadership in groups
- Demonstrating curiosity and open-mindedness

Relationship Skills

Responsible Decision Making
- Recognizing how critical thinking skills are useful both inside & outside of school

Unit Essential Question(s):
- What are the basics of working in the CodeHS environment?
- How does Python differ from Java programming language?
- What are the basic commands of the Python programming language?
- How can I use Python to program robotics?
- What responsibility do programmers have for the consequences of programs they create, whether international or not?

Unit Enduring Understandings:
- Understand how to use CodeHS to learn and write programs
- Understand the basic differences between Java and Python
- Use Python programming to program different robotics.
- Understand the ethical responsibilities that come with creating programs.

Evidence of Learning

Formative Assessments:
- Do Now
- Teacher observations
- Questioning
- Quizzes
- Practice Programs
- Entry tickets
- Exit tickets
- Online games
- Discussions
- Homework

Summative/Benchmark Assessment(s):
- Projects

- Tests
- Chapter Review / Quizzes
- Chapter Assignments
- Labs
- Final Project

Alternative Assessments:
- Portfolio

- Projects
- Online tests / assignments

Resources/Materials:

- Finch Robots - https://www.birdbraintechnologies.com/finch-start teaching/
- http://csunplugged.com/
- www.codehs.com

Key Vocabulary: forward(a_number), backward(a_number), right(degrees), left(degrees), Loop, Indentation, lef(angle), right(angle), speed(number 1-10), Comment, Variable, Function, Function body, Color("red"), Break Down (Decompose), Pseudocode, Decomposition, Top Down Design, input(), Parameters, if Statement, Elif, While loop, For Loop

Lesson Name/Topic	Suggested Pacing Guide Student Learning Objective(s) Suggested Tasks/Activities:		Day(s) to Complete
Introduction to Programming with Turtle Graphics	▪ Define programming/coding and use basic Tracy commands ▪ Locate Tracy on the coordinate plane ▪ Use the penup(), pendown(), and backward() commands in their programs. ▪ Use the left and right commands in order to move Tracy around her grid worlds ▪ Create for loops to repeat code a fixed number of times ▪ Explain when a for loop would be useful tool ▪ Utilize for loops to write programs that would be difficult / impossible without loops. ▪ Using angles inside turning commands ▪ Turn Tracy at angles in conjunction with for loops ▪ Use comments throughout their program ▪ Name elements of code by following specific guidelines and rules in order to create readable and working programs. ▪ Define and call functions ▪ Use commands to draw different shapes and add creativity to programs ▪ Break a large problem down into smaller pieces ▪ Describe what variables are and why they are used in programs.	▪ Students will use the online Coding platform CodeHS to learn Python programming. They will work through lessons, videos and projects. Topics covered in this section are: o Intro to Python with Tracy the Turtle o Tracy's Grid World o Turing Tracy o For Loops o Turning Tracy Using Angles o Comments o Naming Guidelines o Functions o Artistic Effects o Top Down Design o Variables o User Input o Parameters o Using i in For Loops o Extended Loop Control o If Statements o If/Else Statements o While Loops o Putting Together Control Structures ▪ Use Finch Robots and/or other robotics to develop Python programs.	20 Days

- Incorporate user input into their code in order to customize their programs.
- Incorporate parameters into their functions in order to adapt their functions to multiple situations
- Use if and if/else statements in order to have Tracy make decisions.
- Effectively use while loops in their programs.
- Combine control structures to solve complicated problems.
- Choose the proper control structure for a given problem.

Basic Python and Console Interaction
- Print text in Python
- Define Python variables and types
- Incorporate user input into programs
- Convert between variables types.
- Describe the different mathematical operators that can be used in programs.
- Create programs that use basic math to compute useful things
- Create programs that take in user input, do simple computations with the input, and produce useful output.
- Use mathematical operators with Strings.
- Incorporate comments into programs in order to make them more readable.
- Students will use the online Coding platform CodeHS to learn Python programming. They will work through lessons, videos and projects. Topics covered in this section are:
  - o Printing in Python
  - o Variables and Types
  - o User Input
  - o Mathematical Operators
  - o String Operators
  - o Comments
- Use Finch Robots and/or other robotics to develop Python programs.

10 Days

Conditionals
- Create Boolean variable to represent meaningful yes/no values
- Print out the value of a Boolean variable.
- Use if statements for control flow in their programs.
- Explain the meaning of each of the comparison operators.
- Create programs using the comparison operators to compare values.
- Predict the Boolean result of comparing two values
- Describe the meaning and usage of each logical operator.
- Construct logical statements using Boolean variables and logical operators.
- Use floating point numbers
- Round values in progrms.
- Students will use the online Coding platform CodeHS to learn Python programming. They will work through lessons, videos and projects. Topics covered in this section are:
  - o Booleans
  - o If Statements
  - o Comparison Operators
  - o Logical Operators
  - o Floating Point Number and Rounding
- Use Finch Robots and/or other robotics to develop Python programs.

8 Days

Looring	☐ Effectively use while loops in their programs. ☐ Detect and resolve infinite loops. ☐ Implement for loops ☐ Use the variable i as a counter	☐ Students will use the online Coding platform CodeHS to learn Python programming. They will work through lessons,	8 Days

	the flow of control through a program. ☐ Combine control structures to solve complicated problems. ☐ Choose the proper control structure for a given problem. ☐ Modularize programs with functions ☐ Effectively use parameters to customize functions in programs. ☐ Describe the different namespaces with regards to variables and functions. ☐ Remove complexity from their programs by abstracting with functions. ☐ Generalize functions with parameters. ☐ Chain functions together using return values. ☐ Create programs that can gracefully handle	exceptions ☐ Continue to function when an error is raised. videos and projects. Topics covered in this section are: o While Loops o For Loops o Break and Continue o Nested Control Structures ☐ Use Finch Robots and/or other robotics to develop Python programs.	o Functions o Functions and Parameters o Namespaces in Functions o Functions and Return Values o Exceptions ☐ Use Finch Robots and/or other robotics to develop Python programs.
Functions and Exceptions ☐ Control the values of i in a for loop. ☐ Explain the critical difference between break and continue ☐ Describe why a break or continue statement would be needed in coding scenario. ☐ Identify the different control structures that can be used to modify		☐ Students will use the online Coding platform CodeHS to learn Python programming. They will work through lessons, videos and projects. Topics covered in this section are:	6 Days

Strings ■ Use indexing in order to find a specific character in a String	■ Use slicing to select a set of values from a String ■ Explain what immutability is and how it applies to Strings in Python. ■ Iterate over characters in a String using for loops. ■ Use the in keyword to check if a characters is in a String. ■ Use various String methods to alter String values. ■ Students will use the online Coding platform CodeHS to learn Python programming. They will work through lessons, videos and projects. Topics	covered in this section are: o Indexing o Slicing o Immutability o Strings and For Loops o The in Keyword o String Methods ■ Use Finch Robots and/or other robotics to develop Python programs.	7 Days

Creating and Altering Data Structures	■ Create and store information in tuples ■ Explain the characteristics of a tuple. ■ Understand and explain the characteristics of a list	■ Students will use the online Coding platform CodeHS to learn Python programming.	6 Days

Extending Data Structures ■ Use lists to store and recall information. ■ Understand and explain the characteristics of a list ■ Use for loops to go through items in a list.	■ Apply useful list methods to alter and access information about a list. ■ Use 2d lists to store information in rows and columns. ■ Perform list comprehensions in order to alter all items in a list at once. ■ Pack and unpack lists in order to quickly and efficiently assign variables to list items. ■ Use dictionaries to structure data ■ Predict if two values are identical and/or equivalent. They will work through lessons, videos and projects. Topics covered in this section are: o Tuples o Lists o For Loops and Lists o List Methods ■ Use Finch Robots and/or other robotics to develop Python programs. ■ Students will use the online Coding platform CodeHS to learn Python programming. They will work through lessons, videos and projects. Topics covered in this section are: o 2D Lists o List Comprehensions o Packing and Unpacking o Dictionaries o Equivalence vs. Identity ■ Use Finch Robots and/or other robotics to develop Python programs.		

5 Days

- Use all concepts learned in this course to create a word guessing game in Python.
- Final CodeHS Python Project

5 Days

Project: Guess the Word

Additional Resources:

Differentiation/Modification Strategies

Students with Disabilities	English Language Learners	Gifted and Talented Students	Students at Risk	504 Students
• Allow errors • Rephrase questions, directions, and explanations • Allow extended time to answer questions and permit drawing as an explanation • Accept participation on any level, even one word • Consult with	• Assign a buddy, same language or English speaking • Allow errors in speaking • Rephrase questions, directions, and explanations • Allow extended time to answer questions • Accept participation at any level, even one word	• Provide extension activities • Build on students' intrinsic motivation • Consult with parents to accommodate students' interests in completing tasks at their level of engagement	• Provide extended time to complete tasks • Consult with Guidance Counselors and follow I&RS procedures/action plans • Consult with other members of the 7th grade team for specific behavior interventions • Provide	• Allow errors • Rephrase questions, directions, and explanations • Allow extended time to answer questions and permit drawing as an explanation • Accept participation on any level, even one word • Consult with

Case Managers and follow IEP accommodations/ modifications			rewards as necessary	Case Managers and follow IEP accommodations/ modifications

				• Assign a buddy, same language or English speaking

Unit 3- Overview

[redacted]

Core Ideas: In this unit students will work on a final project of their choosing. The project must take into account technology as well as programming. Projects must serve a purpose or help improve something.

Unit 3 - Standards

[redacted]

8.1.12.IC.3 Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.

Test

[redacted]

8.1.12.AP.3 Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.

8.1.12.AP.4 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

8.1.12.AP.5 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

8.1.12.AP.6 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

8.1.12.AP.7 Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.

[redacted]

8.2.12.ED.1 Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.

8.2.12.ED.2 Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.

8.2.12.ED.4 Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.

8.2.12.ETW.3 Identify a complex, global environmental or climate change issue, develop a systemic plan of investigation, and propose an innovative sustainable solution.

8.2.12.EC.2 Assess the positive and negative impacts of emerging technologies on developing countries and evaluate how individuals, non-profit organizations, and governments have responded.

[redacted]

9.2.12.CAP.2 Develop college and career readiness skills by participating in opportunities such as structured learning experiences, apprenticeships, and dual enrollment programs.

9.2.12.CAP.8 Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, and drug tests) used by employers in various industry sectors.

[redacted]

9.4.12.CT.1 Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).

9.4.12.CT.2 Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).

9.4.12.IML.3 Analyze data using tools and models to make valid and reliable claims, or to determine optimal design solutions (e.g., S-ID.B.6a., 8.1.12.DA.5, 7.1.IH.IPRET.8)

9.4.12.TL.1 Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).

[redacted]

9.4.12.TL.4 Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).

Intercultural Statements (Amistad, Holocaust, LGBT, etc...)

LGBTQ and Disabilities NJSA 18A:35- 4.35
Amistad Law NJSA 18A:35- 4.43
Holocaust Law NJSA 18A:35- 28
AAPI Law
NJSA 18A:25- 4.44
Midland Park Public Schools

Explore computer scientist in the LGBTQ community, including but not limited to Sofia Kovalevskaya, Alan Turing, Christopher Strachey, Peter Landin, Edith Windsor, Lynn Conway, Jon Hall, Sphie Wilson, Mary Ann Horton, and Audrey Tang

Explore African-American computer scientist, including [redacted]

but not limited to Clarence Ellis, Melba Roy Mouton, Katherine Johnson, Mary Jackson, and Annie Easley

Explore Jewish computer scientist, including but not limited to Joseph Weizenbaum, Larry Page, Hilary Putnam, Jon von Neumann, Sergery Brin, Saul Amarel and Norbert Wiener

Explore Asian-American/Pacific Islander computer scientist, including but not limited to Peter Tsaai, Nainoa Thompson, Flossie Wong-Staal, Ajay Bhatt, Min Chueh Chang, Roseli Ocamp-Friedmann, Steven Shih Chen, and Ching Wan Tang

Science
HS-ETS1-4 History
6.1.12.SE.14.a
Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem. Explore the various ways women, racial and ethnic minorities, the LGBTQ community, and individuals with disabilities have contributed to the American economy, politics and society

[redacted]

NJSLSA.SL1 Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.

NJSLSA.SL2 Integrate and evaluate information presented in diverse media and formats, including visually, quantitatively, and orally.

[redacted]

Self-Awareness 🎬 Examining prejudices and biases
🎬 Having a growth mindset

Self Awareness Relationship Skills
Management Social

Responsible Decision Making
- Exhibiting self-discipline and self-motivation
- Using planning and organizational skills
- Taking others' perspectives
- Understanding the influences of organizations/systems on behavior

Communicating effectively
- Practicing teamwork and collaborative problem-solving
- Showing leadership in groups
- Demonstrating curiosity and open-mindedness
- Recognizing how critical thinking skills are useful both inside & outside of school

Unit Essential Question(s):
- How can technology be impactful and/or harmful?
- Can I create something that can be impactful?

Unit Enduring Understandings:
- Understand the design process from start to finish
- Understand how feedback drives design
- Understand how technology can have impact on society.

Evidence of Learning

Formative Assessments:
- Do Now
- Teacher observations
- Questioning
- Quizzes
- Practice Programs
- Entry tickets
- Exit tickets
- Online games
- Discussions
- Homework

Summative/Benchmark Assessment(s):
- Projects
- Tests
- Chapter Review / Quizzes
- Chapter Assignments
- Labs
- Final Project

Alternative Assessments:
- Portfolio
- Projects
- Online tests / assignments

Resources/Materials:
- Any technology resources available to students that are in the classroom or that they have at home.

Key Vocabulary:
Engineering process, design, feedback, prototype, testing

Suggested Pacing Guide

Lesson Name/Topic	Student Learning Objective(s)	Suggested Tasks/Activities:	Day(s) to Complete
Concept/ Research	prototype	The topic can be anything but must be approved by the teacher.	come up with several concepts for a prototype
- Research topics for prototype - Come up with a concept for	- Students should research a topic of interest that they can create a prototype for.	- Students should	- Students should get feedback from peers to help determine which

prototype they are going to work on.		2-3 Days	
		needed. 🎬 Take into account scale of model vs. reality. 🎬 Understand what a simulation is in relationship to their prototype 🎬 Design a prototype 🎬 Students will sketch the prototype that they are going to create. They should take into account scale.	🎬 Students should get feedback from peers to help determine design changes that should be made before doing to production. 🎬 Students will physically design their prototype 🎬 Students should get feedback from peers to help determine functionally of the actually prototype
Design 🎬 Sketch a design for their prototype and identify materials that will be			13 Days

Testing 🎬 Test and refine project 🎬 Students will continuously test and refine their prototype to get the functionality desired.
🎬 Students will finalize their project, including a presentation

Teacher Notes: Students should be encouraged to create something helpful or useful. It is alright if the prototype does not work as intended but students should be able to explain why and what they could change if they had more time/resources. Additional Resources:

Differentiation/Modification Strategies

Learners

Midland Park Public Schools

Students with Disabilities

Gifted and Talented Students

Students at Risk 504 Students

English Language

• Allow errors • Rephrase questions, directions, and explanations • Allow extended time to answer questions and permit drawing as an explanation • Accept participation on any level, even one word • Consult with Case Managers and follow IEP accommodations/ modifications	• Assign a buddy, same language or English speaking • Allow errors in speaking • Rephrase questions, directions, and explanations • Allow extended time to answer questions • Accept participation at any level, even one word	• Provide extension activities • Build on students' intrinsic motivation • Consult with parents to accommodate students' interests in completing tasks at their level of engagement	• Provide extended time to complete tasks • Consult with Guidance Counselors and follow I&RS procedures/action plans • Consult with other members of the 7th grade team for specific behavior interventions • Provide rewards as necessary	• Allow errors • Rephrase questions, directions, and explanations • Allow extended time to answer questions and permit drawing as an explanation • Accept participation on any level, even one word • Consult with Case Managers and follow IEP accommodations/ modifications • Assign a buddy, same language or English speaking