Forest Hills School District

COMPUTER SCIENCE

COURSE OF STUDY

Approved by the Forest Hills Board of Education
March 2023

FOREST HILLS SCHOOL DISTRICT

Board of Education

Linda Hausfeld, President
Sara Jonas, Vice President
Bob Bibb, Member
Dr. Leslie Rasmussen, Member
Katie Stewart, Member

District Administrators

Larry Hook, Superintendent
Alana Cropper, Treasurer
Kim Tinsley, Assistant Superintendent
Rob Fellows, Director of Human Resources
Steve Meece, Director of Technology
John Eckert, Director of Business Operations
Betsy Ryan, Director of Special Education/Student Programs
Shane Hartley, Teaching and Learning Director of 7-12 Programs
Bob Buck, Teaching and Learning Director of K-6 Programs

Committee Members

Allyson Petrek, Secondary Learning Design Specialist
Brad Gaertner, Anderson High School
Nate Williams, Secondary Learning Design Specialist
Susan Keeler-Wyborski, Turpin High School

Table of Contents

Introduction

A team of professional, dedicated and knowledgeable 9-12 educators in the Forest Hills School District developed the Computer Science Framework outlined in the document below. This document was created following careful review of the Ohio's Learning Standards for Computer Science (adopted in December 2018) as well as careful attention to current learning theory and best practice in Forest Hills.

It is important to note that the Computer Science Course of Study only applies to the specific Computer Science Courses offered to students in grades 9-12. Students in grades K-6 begin their studies of computer science principles through their Media Course of Study, and students in grades 7-8 explore computer science principles through various STEM (Science, Technology, Engineering, and Math) and Project Lead the Way courses.

Areas of Focus for Computer Science Courses

Computing Systems - Addresses how devices, including hardware and software, interact to accomplish tasks and how students can troubleshoot computing systems when they do not work as intended.

Networks and the Internet - Addresses how networks connect to share information and resources and how students can apply cybersecurity concepts to protect information.

Data and Analysis - Addresses how data can be collected and stored; analyzed and communicated; and used to make more accurate predictions.

Algorithmic Thinking and Programming - Addresses program development, including the use of algorithms, variables, control structures and modules.

Impacts of Computing - Addresses computing's influence on our world by examining the relationship between computing and culture, computing's impact on social interaction, and legal and ethical implications of computing.

Computational Thinking - A problem-solving process that students use to engage with concepts in the computer science standards. This thinking involves formulating problems in a way that can be carried out by a computer. Using computational thinking to solve a problem includes breaking down the problem into manageable parts; recognizing patterns; excluding irrelevant details to abstract or identify general principles that generate these patterns; and developing step - by -step sequences or algorithms to solve the problem and similar problems. Computational thinking can be applied with or without computers, for example, through "unplugged" activities. While computational thinking is a focus in computer science, it also is used in content areas beyond computer science.

**Adapted from the Ohio Standards for Computer Science (Adopted 2018)

Program Philosophy and Goals

Because our age is one of great scientific, technological and social change, education must prepare learners for a future which is unknown and unpredictable. Therefore, we believe that all students should have the opportunity to develop the general range of skills and understanding necessary to function effectively in a society increasingly dependent on computer technology and information.

The computer science program in the Forest Hills School District helps develop our learners into critical thinkers, creative problem solvers, and adaptable learners. The computer science courses within this program empower ALL learners through instruction that:

- Improves logical reasoning and computational thinking by requiring students to recognize patterns, to break problems down into manageable parts, and to think in sequential and analytical methods
- Implements interactive lessons and innovative projects based in real-world applications
- Requires students to collaborate, communicate, and problem solve with others
- Encourages students to take ownership of their learning by independently exploring and analyzing challenging tasks
- Supports inquiry, risk taking, productive struggle, and building resilience

We believe that computer literacy must encompass the following domains, as stated in the Ohio Learning Standards for Computer Science (adopted 2018): computing systems, networks and the internet, data and analysis, algorithmic thinking and programming, and impacts of computing.

Overall, the FHSD computer science program gives students meaningful experiences that allow them to discover and take part in a world that is continually influenced by technology. The ability to use technology strategically is foundational to our students' future success as well as a diverse range of career pathways.

Applied Computer Skills

Theme: This semester course is designed for the student who wants more expertise in using application software for both academic and future-ready pursuits. Students will master a variety of important applications and create personal projects in a collection of current software programs including business software (spreadsheets, forms and presentations), web design (CSS, Javascripts and HTML5). This is the kind of software most likely needed in college, the business world, and social media.

Unit 1: Productivity Software

Students will work with several different productivity software products including Presentations, Spreadsheets, Forms, and Digital Art

Unit Topic Statements/Targets:

In this unit students will learn...

1. The basics of presentation features: links, animations, inserting pictures, transitions, themes, and professionality.
2. To learn and utilize key functions of spreadsheets including functions, data management, statistical analysis, and communicating data trends through charts and graphs.
3. To create and use forms to gather data and information.
4. To use digital software to create drawings and art.

Unit 2: Web Design

This unit will encompass the basics of HTML, proper coding and structure, commenting, hyperlinks, pictures, and styling of websites. Students will learn how to create and publish websites

Unit Topic Statements/Targets:

In this unit students will…

1. Understand the basics of how the internet works and the vocabulary of the web
2. Learn the basics of a HyperText Markup Language
3. Use HTML tags to write a web page
4. Use application software to produce web pages
5. Create multimedia for inclusion on a web page
6. Create web sites and understand hyperlinking
7. Explore scripting languages
8. Create scripts for rollovers, banners, and data validation

Programming

Theme: This introductory course teaches the foundations of computer science and basic programming, with an emphasis on helping students develop logical thinking and problem solving skills.

Unit 1: Introduction to Programming

Students will learn the basic constructs of writing a program, including how to control the flow using conditions and loops.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Commands
2. Defining vs. calling functions/methods
3. Designing functions/methods
4. Program entry points
5. Control flow
6. Looping
7. Conditionals
8. Classes
9. Commenting code
10. Preconditions and postconditions
11. Top down design

Unit 2: Graphics and User Interaction

Students will learn how to call library functions to output graphics.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Variables
2. User input
3. Arithmetic expressions
4. Graphics

Unit 3: Control Structures

Students learn how to use booleans and logical operators with control structures to make more advanced programs.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Boolean values
2. Logical operators
3. Comparison operators

4. For loops
5. Conditionals
6. Nested control structures
7. While loops

Unit 4: Functions and Parameters

Students learn to write reusable code with functions and parameters.

Unit Topic Statements/Targets:

In this unit students will learn….
1. Functions with and without parameters
2. Functions with and without return values
3. Nested control structures
4. Local variables and scope

Unit 5: Animation and Games

Students learn to write reusable code with functions and parameters.

Unit Topic Statements/Targets:

In this unit students will learn….
1. Timers
2. Randomizing games
3. Mouse events
4. Keyboard events

Unit 6: Strings

Students learn to parse and manipulate string values.

Unit Topic Statements/Targets:

In this unit students will learn….
1. Indexing and slicing
2. Math operators on strings
3. For loops over a string
4. String methods

Unit 7: Data Structures

Students learn to write and utilize lists, indices, and list functions/procedures.

Unit Topic Statements/Targets:
In this unit students will learn…. 1. Lists, arrays, tuples 2. For loops and lists 3. List methods

AP Computer Science Principles

Theme: This course introduces students to the foundational concepts of computer science and explores the impact computing and technology have on our society. With a unique focus on creative problem solving and real-world applications, AP Computer Science Principles gives students the opportunity to explore several important topics of computing using their own ideas and creativity, use the power of computing to create artifacts of personal value, and develop an interest in computer science that will foster further endeavors in the field.

Computational Thinking Practices:
The six computational thinking practices represent important aspects of the work that computer scientists engage in:

- Practice P1: Computational Solution Design
  - Design and evaluate computational solutions for a purpose.
- Practice P2: Algorithms and Program Development
  - Develop and implement algorithms.
- Practice P3: Abstraction in Program Development
  - Develop programs that incorporate abstractions.
- Practice P4: Code Analysis
  - Evaluate and test algorithms and programs.
- Practice P5: Computing Innovations
  - Investigate computing innovations.
- Practice P6: Responsible Computing
  - Contribute to an inclusive, safe, collaborative, and ethical computing culture.

Big Ideas:
The five big ideas of the course encompass foundational ideas in the field of computer science.

- **Big Idea 1: Creative Development (CRD)**
  When developing computing innovations, developers can use a formal, iterative design process or experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, collaboration is an important tool to use at any phase of development because considering multiple perspectives allows for improvement of innovations.
- **Big Idea 2: Data (DAT)**
  Data is central to computing innovations because it communicates initial conditions to programs and represents new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of this data. Computers store data digitally, which means that the data must be manipulated in order to be presented in a useful way to the user.
- **Big Idea 3: Algorithms and Programming (AAP)**
  Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems. Using multiple program statements in a specified order, making decisions, and repeating the same process multiple times are the building blocks of programs. Incorporating elements of abstraction, by breaking problems down into interacting pieces, each with their own purpose, makes writing complex programs easier. Programmers need to think algorithmically and use abstraction to define and interpret processes that are used in a program.
- **Big Idea 4: Computing Systems and Networks (CSN)**
  Computer systems and networks are used to transfer data. One of the largest and most commonly used networks is the Internet. Through a series of protocols, the Internet can be used to send and receive information and ideas throughout the world. Transferring and

processing information can be slow when done on a single computer but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.

- **Big Idea 5: Impact of Computing (IOC)**
  Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand how our programs will be used and be responsible for the consequences. As computer users, we need to understand how to protect ourselves and our privacy when using a computer.

Unit 1: Intro to Programming

This course begins with a strong focus on programming in order to allow students to create computational artifacts early on in the course. Students will be able to use their knowledge of programming to explore future topics in the course. Students will learn about the need for programming languages, the uses of programs, how to write programs to solve computational problems, how to design algorithms, how to analyze and compare potential solutions to programming problems, and learn the value and challenges involved in collaborating with others to solve programming problems.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Procedural abstraction and reducing complexity
2. Program documentation
3. Using existing code
4. Commenting code
5. If/else statements
6. Iteration
7. Logic, syntax, and run-time errors
8. Sequencing, selection, and iteration
9. Using existing algorithms
10. Creating new algorithms
11. Optimization and efficiency

Unit 2: Practice PT: Pair-Programming

Students will use language library functions to create a digital image.

Unit Topic Statements/Targets:

In this unit students will….

1. Identify the programming language and purpose of their program.
2. Describe the incremental and iterative development process of their program. How did they divide the program into smaller tasks and make a plan to complete them all?
3. Describe the difficulties and/or opportunities encountered and how they were resolved or incorporated.
4. Identify an algorithm that is fundamental for their program to achieve its intended purpose and includes two or more additional algorithms.

5. Describe how each algorithm within their selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program.
6. Identify an abstraction they developed, and explain how their abstraction helped manage the complexity of their program.

Unit 3: Basics of Programming

This unit introduces students to the basics of a programming language, including variables, user input, control structures, functions with parameters and return values, basic graphics, and how to send messages to objects.

Unit Topic Statements/Targets:

In this unit students will learn….

1. About the variety of programming languages.
2. Pseudocode
3. Variable names
4. Assignment operators
5. Data types
6. Program behavior
7. Testing using inputs
8. Arithmetic expressions
9. Order of operations
10. Modulus
11. String concatenation
12. User input
13. Program output
14. Events

Unit 4: Control Structures

Students learn how to use booleans and logical operators with control structures to make more advanced programs.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Booleans
2. Relational operators
3. Operands
4. Selection
5. Conditional statements
6. Nested conditionals
7. Equivalent boolean statements
8. Random numbers
9. Iteration with loops
10. Different but equivalent algorithms

Unit 5: Functions and Parameters

Students learn to write reusable code with functions and parameters.

Unit Topic Statements/Targets:

In this unit students will learn….

1. User and application input and output
2. Procedures/functions
3. Parameters
4. Return values
5. Using existing algorithms

Unit 6: Basic Data Structures

Students learn to write and utilize lists, indices, and list procedures.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Data values
2. Lists and elements
3. Indices
4. List procedures
5. Data abstraction
6. Translating and transforming data
7. Patterns
8. Extracting and modifying information
9. Traversing a list
10. Iteration statements
11. Search tools
    a. Linear Search
    b. Binary Search
12. Algorithm efficiency
13. Heuristics
14. Simulations and bias
15. Random number generators

Unit 7: Digital Information

Students will learn about the various ways we represent information digitally. Topics covered include number systems, encoding data, programmatically creating pixel images, comparing data

encodings, compressing and encrypting data.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Computing devices
2. Abstraction
3. Program input and output
4. Bits and bytes
5. Overflow errors
6. Range of value limits
7. Binary and decimal systems
8. Lossless data compression
9. Lossy data compression
10. Digital and analog data
11. Decidable problems
12. Heuristic problems
13. Computer viruses
14. Encryption

Unit 8: Practice Performance Tasks and Create Performance Task

Students will practice for the Create Performance Task portion of the AP Exam by implementing a form of cryptography known as Steganography, and will also create their own Image Filter. Then students will complete the Create Performance Task and submit to the AP Digital Portfolio.

Unit Topic Statements/Targets:

In this unit students will….

1. Identify the programming language and purpose of their program.
2. Describe the incremental and iterative development process of their program. How did they divide the program into smaller tasks and make a plan to complete them all?
3. Describe the difficulties and/or opportunities encountered and how they were resolved or incorporated.
4. Identify an algorithm that is fundamental for their program to achieve its intended purpose and includes two or more additional algorithms.
5. Describe how each algorithm within their selected algorithm functions independently, as well as in combination with others, to form a new algorithm that helps to achieve the intended purpose of the program.
6. Identify an abstraction they developed, and explain how their abstraction helped manage the complexity of their program.

Unit 9: The Internet

This unit explores the structure and design of the Internet, and how this design affects the reliability of network communication, the security of data, and personal privacy. Students will learn

about the protocols and algorithms used on the Internet and the importance of cybersecurity.

Unit Topic Statements/Targets:

In this unit students will learn….

1. Protocols
2. Computing devices
3. Computer networks
4. Bandwidth
5. Routing
6. Scalability
7. Fault-Tolerance
8. Redundancy
9. Packets
10. IP, TCP, EDP
11. HTTP
12. Metadata
13. Parallel systems and computing
14. Scalability of systems
15. Distributed computing
16. Efficiency
17. Computing innovations
18. Unintended effects
19. Impacts on society
20. Digital divide
21. Citizen science
22. Crowdsourcing
23. Creative credit and copyright
24. Legal and ethical concerns
25. Identifiable info
26. Digital footprint
27. Authentication
28. Viruses
29. Cybercrime and attacks
30. Encryption

Unit 10: Effects of the Internet

Students will choose an innovation that was enabled by the Internet and explore the positive and negative impacts of their innovation on society, economy, and culture. Students will develop a computational artifact that illustrates, represents, or explains the innovation's purpose, its function, or its effect.

Unit Topic Statements/Targets:

In this unit students will….

1. Explain at least one beneficial effect and at least one harmful effect the Internet-based innovation has had, or has the potential to have, on society, economy, or culture.
2. Identify data privacy, security, or storage concerns for computing innovation.

3. Include citations, as applicable, within their written responses

Unit 11: Data
Students will explore using computational tools to store massive amounts of data, manipulate and visualize data, find patterns in data, and draw conclusions from data. Students will consider how the modern wealth of data collection has impacted society in positive and negative ways.
Unit Topic Statements/Targets:
In this unit students will learn…. 1. Filtering and cleaning data 2. Patterns and trends 3. Search tools 4. Displaying data 5. Combining data sources 6. Metadata 7. Correlation 8. Using a variety of sources 9. Bias 10. Surveys, testing, and interviews

AP Computer Science A

Theme: AP Computer Science A introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language.

Unit 1: Primitive Types

Introduce students to the programming language, focusing on the main method and producing output. Students will learn about primitive data types, create variables, store values and use basic operations on those variables.

Unit Topic Statements/Targets:

In this unit students will learn...

1. Variables and data types
2. Expressions and assignments statements
3. Compound assignment operators
4. User input
5. Casting and ranges of variables

Unit 2: Using Objects

Introduce students to the objects, writing methods, and calling methods.

Unit Topic Statements/Targets:

In this unit students will learn...

1. Objects: Instances of classes
2. Creating and storing objects (Instantiation)
3. Calling a void method
4. Calling a void method with parameters
5. Calling a non-void method
6. String objects: concatenation, literals, and more
7. String methods
8. Wrapper classes: Integer and Double
9. Using the math class

Unit 3: Boolean Expressions and if Statements

Explore conditional statements, boolean expressions, and equivalent statements.

Unit Topic Statements/Targets:

In this unit students will learn...

1. Boolean expressions
2. If statements and control flow
3. If-else statements
4. Else-if statements
5. Compound boolean expressions
6. Equivalent boolean expressions
7. Comparing objects

Unit 4: Iteration

Using while and for loops

Unit Topic Statements/Targets:

In this unit students will learn...

1. While loops
2. For loops
3. Developing algorithms using strings
4. Nested iteration
5. Informal code analysis

Unit 5: Writing Classes

Constructors, Methods, Scope and Access, Static Variables and Methods

Unit Topic Statements/Targets:

In this unit students will learn...

1. Anatomy of a class
2. Constructors
3. Documentation with comments
4. Accessor methods and mutator methods
5. Writing methods
6. Static variables and methods
7. Scope and access
8. This keyword
9. Ethical and social implications of computing systems

Unit 6: Arrays

Create/traverse arrays, algorithms using arrays

Unit Topic Statements/Targets:

In this unit students will learn...

1. Array creation and access
2. Traversing arrays
3. Enhanced for loop for arrays
4. Developing algorithms using arrays

Unit 7: ArrayLists

Create/traverse arraylists, algorithms using arraylists

Unit Topic Statements/Targets:

In this unit students will learn...

1. Introduction to ArrayList
2. ArrayList methods
3. Traversing ArrayLists
4. Developing algorithms using ArrayLists
5. Searching and sorting
6. Ethical issues around data collection

Unit 8: 2D Arrays

Create/traverse 2D arrays, algorithms using 2D arrays

Unit Topic Statements/Targets:

In this unit students will learn...

1. 2D arrays
2. Traversing 2D arrays

Unit 9: Inheritance

Creating classes, superclasses, and subclasses; creating and calling methods, overriding methods, creating references and hierarchies; polymorphism

Unit Topic Statements/Targets:

In this unit students will learn...

1. Creating subclasses and superclasses
2. Writing constructors for subclasses
3. Overriding methods
4. Super keyword
5. Creating references using inheritance hierarchies

6. Polymorphism
7. Object superclass

Unit 10: Recursion

Utilize recursion in creating and calling methods; recursive searching and sorting

Unit Topic Statements/Targets:

In this unit students will learn...

1. Recursion
2. Recursion searching and sorting

Advanced Studies in Computer Science

Theme: This course will be tailored to meet the needs of the students. The student should be self-motivated and will be responsible for the development of the tasks and the assessments. Students will suggest a topic they want to study, and may include learning a new programming language, Robotics, Virtual Reality, Advanced Web Design, Java GUI Applets, Gaming, Advanced Data Structures, Artificial Intelligence, etc.

There is no specific syllabus for this course since each student may choose their own learning resources and goals

Resource Materials

Grade/Course	Resource	Publisher
AP Computer Science Principles	AP Central: AP Computer Science Principles Curriculum	AP Central
AP Computer Science A	AP Central: AP Computer Science A Curriculum	AP Central
All Computer Science Courses	CodeHS Computer Science Curriculum (Available online through the CodeHS website)	CodeHS