CARY ACADEMY

UPPER SCHOOL
COMPUTER SCIENCE CURRICULUM

Address

Cary Academy
1500 N. Harrison Ave.
Cary, NC 27511

Phone

Administration: (919) 677-3873
Middle School: (919) 228-4600
Upper School: (919) 228-4544

Online

Email: info@caryacademy.org
Website: www.caryacademy.org

CARY ACADEMY

Upper School Curriculum
Computer Science: CSP 101: Introduction to Programming

CSP 105: INTRODUCTION TO PROGRAMMING

This course introduces students to basic programming structures and skills as they relate to the object-oriented programming languages. This course will emphasize a strong ability to utilize basic programming control structures and data types while exposing student to more advanced programming concepts. The focus language for this class is C#.

GENERAL COURSE OBJECTIVES:

- Introduce students to programming using C#
- Recognize and implement good development habits: start with an algorithm, program incrementally to test and troubleshoot as programs are developed.
- Expose students to software applications written for text console, windows desktop and web browser.
- Allow for future growth at an individual level, especially for those who have mastered a current topic of study.

GENERAL COURSE OUTLINE:

UNIT	CONCEPTS
History of Electronic Computing	Quick introduction to the history of electronic computing. From mechanical tabulators to electro-mechanical special purpose computational devices to modern electronic computers
Algorithm and Flowcharts basics	Introduction Visualizing problems using decision trees and tables
C# Basics: Variable, Types and Operators	Introduction to C# language Variables Types Declaration Initialization Math and Logical Operators
Program Flow Control	if/else, switch and for, for each, while and do-while loops
Introduction to Methods and Functions	Methods and Functions Code re-use Arguments, Parameters and Return Values. Overloaded Methods
Arrays	Basics of Arrays and Collections Storing and Sorting Two-Dimensional Arrays Nested Loops
Introduction to Windows Desktop Development	Basics of Windows Forms Event driven programming Using Windows Controls.
Classes and Objects	Object-oriented Programming Concepts

CARY ACADEMY

	Understanding of Classes and Objects. Inheritance and Polymorphism
Introduction to Web Development	Basics of Web Development HTML 5 CSS Java Script Using C# developing simple Web Apps

CARY ACADEMY

Upper School Curriculum
Computer Science: CSP155: Computer and Network Essentials

CSP 105: INTRODUCTION TO PROGRAMMING

This course introduces students to basic programming structures and skills as they relate to the object-oriented programming languages. This course will emphasize a strong ability to utilize basic programming control structures and data types while exposing student to more advanced programming concepts. The focus language for this class is C#.

GENERAL COURSE OBJECTIVES:

Have the core knowledge base to take the CompTIA A+ Certification tests

- There are two tests for the CompTIA A+ certification
- **220-901**
  Covers PC hardware and peripherals, mobile device hardware, networking and troubleshooting hardware and network connectivity issues. (https://certification.comptia.org/certifications/a)
- **220-902**
  Covers installing and configuring operating systems including Windows, iOS, Android, Apple OS X and Linux. It also addresses security, the fundamentals of cloud computing and operational procedures. (https://certification.comptia.org/certifications/a)

Learn various problem formulation and problem solving skills.

- Material used to achieve this goal will be based on the theories, literature and practices of Dr. David Jonassen from the University of Missouri and Douglas Lecorchick and Dr. Matthew Lammi of North Carolina State University.
- The goal is to employ the various theories, literature and practice to observe students' responses. After observing and analyzing how students learn problem formulation and problem solving, we will continue to redesign the process to see what the students feel helped them achieve overall success.
- In combining and redesigning the problem formulation and problem solving skills, we hope students will be able decide for their own learning what works best for them. Students should not be stuck using one model when various models or a combination of models might work best for their own learning.
- This could be time consuming for the instructors teaching this curriculum. However, through various labs and observations instructors should find this goal achievable. Instructors should anticipate to learn their own problem formulation and problem solving skills to help students achieve a more critical thinking process.

Learn various design thinking processes, to then apply them to problem formulation.

- Three design thinking models will be taught, so students can apply solutions to the problems that they might encounter in a lab setting or in real world experiences.
- The three models to be used will the IDEO, Stanford d.School and the NASA Engineering Design Process.

STANFORD D.SCHOOL	IDEO	NASA ENGINEERING DESIGN PROCESS
Empathize	Discovery	Identify the Problem
Define	Interpretation	Identify Criteria and Constraints
Ideate	Ideation	Brainstorm Possible Solutions
Prototype	Experimentation	Generate Ideas
Test	Evolution	Explore Possibilities

		Select an Approach
		Build a Model or Prototype
		Refine the Design

- In combining and redesigning the processes above, we hope students will be able decide for their own learning what works best for them. Students should not be stuck using one model when various models or a combination of models might work best for their own learning.

In learning the above overall goals, the students' critical thinking skills will be demonstrated through various lab settings and problem based learning scenarios in collaborative groups. Many problems encountered outside the classroom take a collaborative effort and students must learn that they need to rely on the thoughts and ideas of others to help solve a problem.

SCOPE AND SEQUENCE:

Computer and Network Engineering (CANE) Class will have the scope and sequence:

UNIT	TARGETED SKILLS
Knowledge to Pass the CompTIA A+ Certification Tests	o Work on labs to apply different Design Thinking models. o Compare the design thinking models and see if they can make their own or use one that already exists. o Design Thinking model they come up with will then be used throughout the course for the year. o Have students work on computer issues to come up with a problem formulation and compare that to the Comp TIA A+ objectives. o Analysis different problem formulation models and see what works best for that student. o Come up with a problem formulation map to use for course throughout the year. o Above skills will be assessed by the instructor through the various collaborative labs and individual problem formulation scenarios.
Learn Design Thinking and Problem Formulation skills	o Acquire the knowledge to build a computer on their own or as a group and identify the various parts needed to build the computer. o Work on what the computer design will be and what parts they need to build the computer. o Price out various computers to compare a high, medium and low cost computers. o Work on a budget set by the instructor to build a computer and order the parts needed for that computer. o Figure out what operating system to install on the computer and install the operating system on their own or as a group.
Learn to build a computer and how the hardware and software interacts	o Look various coding languages. o Learn basic programming structure. o Apply programing structure to a problem and use the computer they built to code with. o Students will have a problem set or real world example to solve with basic command programing.
Learn Basic Command Prompts and Coding	o Learn the different networking topologies. o Setup a closed network and have several computers networked together. o Make their own network cables to have the computers talk to each other. o Setup a basic router to have computers networked.
Networking	o Build a robot that will apply the above units and skill sets. o Use Design Thinking and Problem formulation skills to have a set plan in place for building the robot. o Build circuitry and other components so the robot can have an onboard computer.

	o Use basic programming to have the robot do certain objectives.

CARY ACADEMY

Upper School Curriculum
Computer Science: CSP201: Intermediate Programming

This course will delve fully into the Java programming language. Student's will continue to build on the skills learned in the Intro to Programming courses while continuing to learn about advanced data structures, algorithm designs, and object-oriented programming. Prerequisite: CPS105: Intro to Programming.

GENERAL COURSE OBJECTIVES:

- Reinforce and strengthen fluency of concepts acquired in CPS105: Intro to Programming.
- Recognize and implement good development habits: start with an algorithm, program incrementally to test and troubleshoot as programs are developed.
- Encourage individual exploration of other languages or topics during times such as Computer Science Week or in breaks between units.
- Allow for growth at an individual level, especially for those who have mastered a current topic of study.

GENERAL COURSE OUTLINE

UNIT	TARGETED SKILLS
Java Basics	o Input and output in Java o Variables and arithmetic operators o Boolean variables and control structures
Java Prebuilt Methods	o Math Class o Formatting output o Random numbers o Character Class o String Class o GUI Basics
Text Files	o Creating and writing to text files o Reading from a text file o Appending to a file o File existence o String class and files
Java Methods	o Increasing efficiency and elegance o Passing parameters o Return methods
Arrays	o Creation & navigation o Copying & manipulation o Searching o Sorting o Passing Arrays
Two-dimensional Arrays	o Setup & Syntax o Navigation & manipulation with nested loops
Object Oriented Programming (OOP)	o Creating classes o Constructors o Creating Objects o Overloaded methods o Class variables o Class methods

CARY ACADEMY

Intermediate OOP	o Encapsulation o Design considerations o Instance variables and using 'this'
Arrays and Classes	o Using arrays with and in classes o Arrays of objects o For each loop
ArrayLists	o ArrayList class o ArrayLists versus standard arrays o LinkedList class
Recursion	o Introduction to theory o Tracing examples of recursion o Comparing recursive and iterative solutions
Advanced OOP	o Inheritance o Polymorphism o Abstract methods o Interfaces
GUI Programming Basics	o Event driven programming basics o JFrame class o Java components o Listeners o Inner classes o Layout managers

CARY ACADEMY

Upper School Curriculum
Computer Science: CPS303: E-imacs Ap Computer Science

Students enrolled in ADV Computer Science at Cary Academy will take a course online through the Institute for Mathematics & Computer Science. From their course catalog: The course can be completed in eight months, allowing students time to review for and take the AP exam. Not only does the course cover all the contents required for the AP exam, but it also contains optional sections covering more advanced topics. Interspersed within a well-organized exposition are exercises to be completed using an embedded Java compiler, graded coding activities, eight labs, and graded tests. Prerequisites: CPS201: Intermediate Programming.

GENERAL COURSE OUTLINE

UNIT	TARGETED SKILLS
Java Basics	o Variable and Expressions o Program Control o Methods
Object-oriented Programming	o Object-oriented Programming Concepts o Simple Objects o Inheritance and Polymorphism o Class Definitions Revisited o Abstractions
Algorithms	o Introduction o Searching and Sorting o Program Analysis
Advanced Topics	o Introduction o Searching and Sorting o Program Analysis o Data Structures
Java Basics	o Variable and Expressions o Program Control o Methods

CARY ACADEMY

Upper School Curriculum
Computer Science: CPS405:
Advance Studies In Computer Science Adv

Structured similarly to an independent study, Advanced Studies in Computer Programming provides student the opportunity to further their knowledge in a topic agreed upon by the student and IS sponsor. Topics may focus on learning different programming languages, computer hardware, network infrastructure, or application development. Students may also choose to pursue an industry recognized certification from Microsoft, Cisco, CompTIA, and more. Meeting times will work around a student's class schedule and may contain an online component dependent of topic of study.

GENERAL COURSE OBJECTIVES

- Design a course of study based on individual interests within computer science.
- Develop a year-long or trimester-long project plan(s) based on the individual's proposed area of study.
- Schedule a set of manageable goals and define success for themselves.
- Critique and defend individual projects based on goals and success markers.
- Utilize design and computational thinking as guides for the creation and evaluation process.

GENERAL COURSE OUTLINE:

TRIMESTER 1	TRIMESTER 2	TRIMESTER 3
o Identify areas of interest. o Research topics of interest. o Define project and timeline. o Set goals and determine definition of success. o Design and implement according to timeline. o Present and defend project for critique.	o Translate critique into actionable items. o Differentiate and select items to be examined further. o Edit timeline, goals, and definition of success. o Design and implement according to timeline. o Present and defend project for critique.	o Translate critique into actionable items. o Differentiate and select items to be examined further. o Edit timeline, goals, and definition of success. o Design and implement according to timeline. o Present and defend project for critique.