

SCHOOL DISTRICT OF THE CHATHAMS

AP Computer Science A Grades 10 - 12 Full Year

Course Overview

The work in this course is devoted to programming in the language of Java, and is presented at a typical first-year undergraduate level. The requirements are exceptionally demanding with emphasis on programming, methodology, algorithms, classes, and stream properties. Course content follows the College Board's Advanced Placement curriculum for the AP test in Computer Science A.

New Jersey Student Learning Standards

The New Jersey Student Learning Standards (NJSLS) can be located at www.nj.gov/education/cccs/2020/.

Computing Systems

8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.

8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

Impacts of Computing

8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

8.1.12.IC.2: Test and refine computational artifacts to reduce bias and equity deficits.

Data & Analysis

8.1.12.DA.1: Create interactive data visualizations using software tools to help others better understand real world phenomena, including climate change.

8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.

8.1.12.DA.6: Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.

Algorithms & Programming

8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.

8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.

8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify tradeoffs 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.

8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.

8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.

Technology Standards

9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.

21st Century Integration | NJSLS 9

9.4.12.CI.1: Demonstrate the ability to reflect, analyze and use creative skills and ideas.

9.4.12.CI.2: Identify career pathways that highlight personal talents, skills and abilities.

9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement and transition.

9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice.

9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

Career Ready Practices

CRP2. Apply appropriate academic and technical skills

CRP4. Communicate clearly and effectively and with reason.

CRP6. Demonstrate creativity and innovation.

CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.

CRP11. Use technology to enhance productivity.

CRP12. Work productively in teams while using cultural global competence.

Interdisciplinary Connections

Science

- HS-ETS1-4 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.

Units of Study

Unit 1: Course Introduction & Review (~27 days)

- What is the hierarchical structure of a library?
- How do we import packages/classes?
- What are some packages available in the JSL?
- How do we implement the scanner class?
- How do we implement the random class?
- How do we implement the math class?
- What does the 'static' keyword mean?
- How do we implement the decimalformat class?
- What are the equality/relational operators?
- What are the logical operators?

Revision Date: November 2021

- What is short-circuiting?
- How do we use logical operators in conjunction with conditionals to alter flow of control?
- Nesting revisited
- What are the issues with dealing with doubles in java?
- How do we create a tolerance?
- How do we compare char vs comparing Strings?
- How do we implement a switch statement?
- How does a Do statement differ from a While and how do we implement one?
- How do we implement a For statement?
- How do we trace nested For loops?
- How do we determine what kind of loop is most appropriate to use?
- What is the purpose of a Try and how do we implement it?
- How do we access data from a txt file?
- What is PrintWriter and how do we write to a text file?
- What is the meaning of Throws?

Unit 2: 1D and 2D Arrays (~15 days)

- How can large amounts of data be stored in a computer efficiently?
- How can data be organized and retrieved in data structures?

Unit 3: Objects (~35 days)

- What is an object?
- What is object oriented programming and why is it useful?
- Why are classes so important in object oriented programming?

Unit 4: Data Structures (~35 days)

- How can large amounts of data be stored in a computer efficiently?
- How can data be organized and retrieved in data structures?

Unit 5: Inheritance and Polymorphism (~35 days)

- What is polymorphism?
- How does polymorphism relate to inheritance?
- What would object oriented programming be like without inheritance?

<h3>Learning Objectives/Discipline Standards of Practice</h3>

Learning Objectives:

- Understand the structure of a Library.
- Understand the function of the Scanner class and methods available.
- Understand the function of the Random class and methods available.
- Understand the function of the Math class and methods available.
- Distinguish between static and non-static methods.
- Understand the function of the DecimalFormat class and methods available.
- Write programs which implement the aforementioned classes.
- Revisit the concept of If and While, singularly and nested.
- Understand the various equality, relational and logical operators.
- Understanding how to alter flow of control.

- Revisit comparisons and issues (binary representation and comparing doubles in Java, comparing char vs Strings).
- Write programs implementing boolean logic and switch statements.
- Understand the difference between a While, Do and For loop.
- Understand the structure of a For loop.
- Navigate the flow of nested For loops.
- Understand the purpose and function of Try/Catch.
- Introduce reading from and writing to a text file.
- Introduce the concept of Throws.
- Declare and initialize 1D and 2D arrays.
- Populate and transverse using loops and enhanced loops.
- Insert and Delete data from arrays.
- Prevent IndexOutOfBounds Exceptions.
- Pass arrays as parameters.
- Use instance fields and local variables appropriately.
- Create constructors and methods with parameters.
- Use public and private declarations appropriately.
- Create objects and aliases.
- Compare objects using == and .equals.
- Create and use static methods and variables.
- Understand nullPointers.
- Use Wrapper classes with primitive data types.
- Declare and initialize arrays and ArrayLists.
- Transverse using loops and enhanced loops.
- Insert and Delete data from arrays and ArrayLists.
- Prevent IndexOutOfBounds Exceptions.
- Identify and trace binary and sequential search algorithms.
- Identify and trace insertion, selection and merge sort algorithms.
- Select appropriate algorithms in different situations.
- Design and write sub-classes and super-classes.
- Use type compatibility and casting appropriately.
- Design and use polymorphic methods.
- Design and use overloaded and overridden methods.
- Design and use interfaces and abstract classes.

Discipline Standards of Practice:

- Computing Systems
 - People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form.
- Networks and the Internet
 - Computing devices typically do not operate in isolation. Networks connect computing devices to share information and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world.
- Impacts of Computing
 - Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their

behaviors and cultural and social interactions, and, in turn, computing influences new cultural practices.

- Data & Analysis
 - Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.
- Algorithms & Programming
 - An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems.
- Engineering Design
 - People design for enjoyment and to solve problems, extend human capabilities, satisfy needs and wants, and improve the human condition. Engineering Design, a systematic approach to creating solutions to technological problems and finding ways to meet people's needs and desires, allows for the effective and efficient development of products and systems.
- Interaction of Technology and Humans
 - Societies influence technological development. Societies are characterized by common elements such as shared values, differentiated roles, and cultural norms, as well as by entities such as community institutions, organizations, and businesses. Interaction of Technology and Humans concerns the ways society drives the improvement and creation of new technologies, and how technologies both serve and change society.
- Nature of Technology
 - Human population, patterns and movement focus on the size, composition, distribution, and movement of human populations and how they are fundamental and active features on Earth's surface. This includes understanding that the expansion and redistribution of the human population affects patterns of settlement, environmental changes, and resource use. Patterns and movements of population also relate to physical phenomena including climate variability, landforms, and locations of various natural hazards and their effects on population size, composition, and distribution.
- Effects of Technology on the Natural World
 - Many of engineering and technology's impacts on society and the environment are widely regarded as desirable. However, other impacts are regarded as less desirable. Effects of Technology on the Natural World concerns the positive and negative ways that technologies affect the natural world.
- Ethics & Culture
 - Ethics and Culture concerns the profound effects that technologies have on people, how those effects can widen or narrow disparities, and the responsibility that people have for the societal consequences of their technological decisions.

Instructional Resources and Materials

Whole class resources have been identified with an asterisk.

Resources

- *AP CSA Java* by Runestone Academy (edited by Beryl Hoffman)

Revision Date: November 2021

Materials

- College Board provided materials including
 - AP labs Magpie
 - PictureLab

Assessment Strategies

Assessment is designed to measure a student's mastery of a course standard and learning objective. Assessment can be used for both instructional purposes (formative assessment) and for evaluative purposes (summative assessment).

The following is a general list of the many forms assessment may take in learning.

- Tests
- Quizzes
- Projects
- Unit Assessments

Course Specific Assessments Include:

- AP Labs