CAESAR CIPHER LESSON

Sean Gallop – Colorado Academy, Boulder, CO

This material is based upon work supported by
the **National Science Foundation** under Grant No.1548315.

Additional materials may be found at www.ncyte.net

CAESAR CIPHER LESSON

Caesar Ciphers are perhaps the simplest encryption protocol, and for this reason, are a great place for high school students to engage with cryptography for the first time. In this Cybersecurity Concept Lesson (CCL), students learn about the purpose of cryptography through all of the following: a) an investigation of Caesar Ciphers' purpose, historical context, and strengths and weaknesses, b) hands-on, in-class activities, c) teacher-led discussion, and d) a multi-day programming assignment. The programming assignment both advances students' understanding of the nature and purpose of Caesar Ciphers, and students' programming ability through engagement with a substantive problem.

OVERVIEW

Prerequisite Knowledge: Students need to be familiar with the following programming fundamentals: conditional statements, iteration, arrays, array processing, strings, modulus mathematics, and string processing. This CCL assumes that students have been introduced to the general topics of cybersecurity and cryptography.

Length of Completion: The CCL is designed to take approximately 100-150 minutes. A programming assignment is required to be completed outside of class.

Homework: Students should read the *Wikipedia* article on Caesar Ciphers prior to the start of Day 1.

Students should watch the *Khan Academy* video on Caesar Ciphers prior to the start of Day 2. Students will submit a programming assignment on the day after the end of the CCL.

Learning Setting: The CCL assumes that students are able to move around and work in small groups.

Lab Environment: While the CCL does not specify a particular programming language, it assumes that students have in-class access to the Internet and an IDE they are familiar with.

Activity/Lab Tasks: The CCL includes five separate activities: two in-class, small-group, hands-on activities, one in-class, Internet-based activity, one summative assessment, and one multi-day programming assignment. The following files are required for the activities:

- 02.CaesarCipher_Presentation.pptx
- 03.CaesarCipher_InClass_EncryptionActivity.docx
- 04.CaesarCipher_InClass_ScavengerHuntActivity.docx
- 05.CaesarCipher_InClass_ScavengerHuntActivity_Solutions.docx
- 06.CaesarCipher_InClass_DecryptionActivity.docx
- 07.CaesarCipher_InClass_DecryptionActivity_Solutions.docx

08.Programming Folder

08.Processing Folder

- CaesarCipher_JavaProcessingArrayList.docx
- CaesarCipher_SimpleFullProcessingCode.docx
- CaesarCipher_SimpleStarterProcessingCode.docx

08.Pseudocode Folder

- CaesarCipher_PseudoCodeFullAnswersArray.docx
- CaesarCipher_PseudoCodeFullAnswers.docx
- CaesarCipher_PseudoCodeReference.docx

08.Python Folder

- CaesarCipher_FullCodeList.docx
- CaesarCipher_SimpleFullPythonCode.docx
- CaesarCipher_SimpleStarterProcessingCode.docx

LEARNING OBJECTIVES AND AP CSP ALIGNMENT

LESSON LEARNING OBJECTIVES

Students will:

1) Investigate how Caesar Ciphers work to achieve the goal of confidentiality of information, their historical context, and strengths and weaknesses,

2) Encrypt and decrypt text using Caesar Ciphers,

3) Implement a Caesar Cipher in a programming language, and

4) Ascertain the correctness of their program as a functional program to protect information assets and computing resources.

ASSOCIATED AP CSP SUB LEARNING OBJECTIVES

AP Computer Science Principles Course, Big Idea 1: Creative Development

- LO CRD-1.A Explain how computing innovations are improved through collaboration.
    - CRD-1.A.3 Effective collaboration produces a computing innovation that reflects the diversity of talents and perspectives of those who designed it.
- LO CRD-1.C Demonstrate effective interpersonal skills during collaboration.
    - CRD-1.C.1 Effective collaborative teams practice interpersonal skills, including but not limited to: communication, consensus building, conflict resolution, and negotiation.
- LO CRD-2.B Explain how a program or code segment functions.
    - CRD-2.B.1 A program is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as software.
- LO CRD-2.I For errors in an algorithm or program:
        - Identify the error.
        - Correct the error.
    - CRD-2.I.1 A logic error is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly.
    - CRD-2.I.2 A syntax error is a mistake in the program where the rules of the programming language are not followed.
    - CRD-2.I.3 A run-time error is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors.

AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 3: ALGORITHMS AND PROGRAMMING

- LO AAP-1.A Represent a value with a variable.
    - AAP-1.A.1 A variable is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values.
    - AAP-1.A.2 Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables.

    - AAP-1.A.3 Some programming languages provide types to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings.
    - AAP-1.A.4 Some values are better suited to representation using one type of data rather than another.
- LO AAP-1.C Represent a list or string using a variable.
    - AAP-1.C.1 A list is an ordered sequence of elements. For example, [value1, value2, value3, …] describes a list where value1 is the first element, value2 is the second element, value3 is the third element, and so on.
    - AAP-1.C.3 An index is a common method for referencing the elements in a list or string using natural numbers.
    - AAP-1.C.4 A string is an ordered sequence of characters.
- LO AAP-2.C Evaluate expressions that use arithmetic operators.
    - AAP-2.A.1 An algorithm is a finite set of instructions that accomplish a specific task.
    - AAP-2.A.3 Algorithms executed by programs are implemented using programming languages.
    - AAP-2.C.1 Arithmetic operators are part of most programming languages and include addition, subtraction, multiplication, division, and modulus operators.
    - AAP-2.C.2 The exam reference sheet provides a MOD b, which evaluates to the remainder when a is divided by b. Assume that a is an integer greater than or equal to 0 and b is an integer greater than 0. For example, 17 MOD 5 evaluates to 2.
- LO AAP-2.D Evaluate expressions that manipulate strings.
    - AAP-2.D.1 String concatenation joins together two or more strings end-to-end to make a new string.
- LO AAP-2.G Express an algorithm that uses selection without using a programming language.
    - AAP-2.G.1 Selection determines which parts of an algorithm are executed based on a condition being true or false
- LO AAP-2.H For selection:
        - Write conditional statements.
        - Determine the result of conditional statements.
    - AAP-2.H.1 Conditional statements or "if-statements" affect the sequential flow of control by executing different statements based on the value of a Boolean expression.

- LO AAP-2.K For iteration:
        - Write iteration statements.
        - Determine the result or side-effect of iteration statements.
    - AAP-2.K.1 Iteration statements change the sequential flow of control by repeating a set of statements zero or more times, until a stopping condition is met.
- LO AAP-2.M For algorithms:
        - Create algorithms.
        - Combine and modify existing algorithms.
    - AAP-2.M.1 Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms.
- LO AAP-2.N For list operations:
        - Write expressions that use list indexing and list procedures.
        - Evaluate expressions that use list indexing and list procedures.
    - AAP-2.N.1 The exam reference sheet provides basic operations on lists, including: accessing an element by index, assigning a value of an element of a list to a variable, assigning a value to an element of a list, inserting elements at a given index, adding elements to the end of the list, removing elements, and determining the length of a list.
- LO AAP-2.O For algorithms involving elements of a list:
        - Write iteration statements to traverse a list.
        - Determine the result of an algorithm that includes list traversals.
    - AAP-2.O.1Traversing a list can be a complete traversal, where all elements in the list are accessed, or a partial traversal, where only a portion of elements are accessed.
    - AAP-2.O.2 Iteration statements can be used to traverse a list.
- LO AAP-3.A For procedure calls:
        - Write statements to call procedures.
        - Determine the result or effect of a procedure call.
    - AAP-3.A.1 A procedure is a named group of programming instructions that may have parameters and return values.
    - AAP-3.A.2 Procedures are referred to by different names, such as method or function, depending on the programming language.

    - AAP-3.A.3 Parameters are input variables of a procedure. Arguments specify the values of the parameters when a procedure is called.
    - AAP-3.A.4 A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has executed, flow of control is returned to the point immediately following where the procedure was called.
- LO AAP-3.B Explain how the use of procedural abstraction manages complexity in a program.
    - AAP-3.B.1 One common type of abstraction is procedural abstraction, which provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it.
    - AAP-3.B.2 Procedural abstraction allows a solution to a large problem to be based on the solution of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems.
    - AAP-3.B.5 Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments.
- LO AAP-3.E For generating random values:
        - Write expressions to generate possible values.
        - Evaluate expressions to determine the possible results
    - AAP-3.E.1 The exam reference sheet provides RANDOM(a, b) which generates and returns a random integer from a to b, inclusive. Each result is equally likely to occur. For example, RANDOM (1, 3) could return 1, 2, or 3.
    - AAP-3.E.2 Using random number generation in a program means each execution may produce a different result.

AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 5: IMPACT OF COMPUTING

- LO IOC-2.C Explain how unauthorized access to computing resources is gained.
    - IOC-2.C.1 Phishing is a technique that attempts to trick a user into providing personal information. That personal information can then

be used to access sensitive online resources, such as bank accounts and emails.

LESSON DETAILS

Overview of Lessons: The CCL is broken down into two 60-minute lessons taught on consecutive days.

Day 1:

- Teacher-Led Discussion (20 minutes)
- In-Class Encryption Activity (15 minutes)
- In-Class Internet Scavenger Hunt (15 minutes)
- Start of Programming (10 minutes)

Day 2:

- Continue Programming (10 minutes)
- In-Class Decryption Activity (30 minutes)

	DAY 1				DAY 2
Teacher-led Discussion on Caesar Cipher, Ciphers, Cryptography, and Programming	In-Class Encryption Activity	In-Class Internet Activity	Start Programming	Continue Programming	In-Class Decryption Activity
	Programming	Assignment:	Encryption	and	Decryption

DAY 1

Teacher-Led Discussion - 20 Minutes

02.CaesarCipher_Presentation.pptx

The teacher should present and lead the discussion using the slides. To the extent possible, the teacher should reference the Khan Academy video on Caesar Ciphers that students watched as homework for the first day of the CCL. By connecting the previous night's homework in this manner, students hopefully feel better prepared for this new material and more comfortable raising issues and participating in the teacher-led discussion.

Students gain an understanding of Caesar Ciphers, their historical context and their strengths and weaknesses, and how Frequency Analysis can be used to easily compromise any Caesar Cipher.

In-Class Encryption Activity - 15 minutes
03.CaesarCipher_InClass_EncryptionActivity.docx

This activity should be done in groups of four students. Each student in the group writes down three brief (less than 25 letters) lyrics from three favorite songs (Section A). Students write in plaintext in the spaces provided, using all capital letters and no punctuation (except for blank spaces). Using a Caesar Cipher shift of their own choosing, students then encrypt their three lyrics into three separate lines of ciphertext (Section B). Next, students carefully fold over the top of the sheet so that the contents of section A are no longer visible. Once done, each student should hand their activity sheet to the student on their right.

The student receiving the sheet then attempts to identify one lyric. Students are given five minutes to decrypt one encrypted lyric on the sheet and guess the song it came from (Section C) and then be given two additional minutes to write one or two sentences about their decryption success, partial success, or failure (Section D). Students again pass the sheet to the next student to the right. The activity continues until all three students in the group have attempted to decrypt three lyrics.

The students learn to encrypt text using Caesar Ciphers, and decrypt text that has been coded using Caesar Ciphers.

In-Class Internet Scavenger Hunt - 15 minutes
04.CaesarCipher_InClass_ScavengerHuntActivity.docx
05.CaesarCipher_InClass_ScavengerHuntActivity_Solutions.docx

The class is divided into groups of three or four. The Scavenger Hunt Activity sheet contains a Caesar Cipher in section A. The cipher was created with a shift between one and three (inclusive). Students should decrypt that cipher because it is a clue to where the next cipher can be found. Once the cipher in section A has been decrypted, students should write down where they believe the next cipher can be found. Once all the groups have finished deciphering the clue and written down where they each think the next cipher can be found, the teacher reads off the answers and then reveals the location of the second cipher. Groups who got the answer correct for section A receive one point. The first, second, and third place correct finishers receive three, two, and one point extra (respectively).

The process begins again for the second cipher with the same resolution process and scoring.

Once complete, the process begins again for the third cipher. This last cipher reveals a clue to the location of a hidden rubber duck. The winner of the overall completion receives the duck.

Regarding rules, students can work together in any arrangement they wish. Students may use the Caesar Cipher tool in the Black Chamber website for all their decryption work:

Plaintext clues are as follows:

First Clue: DONT THROW AWAY A DUCK (next clue in trash can)

Second Clue: QUACK BEHIND THE TIME (next clue behind the clock)

Last Clue: DUCK DOWN TO MY DESK (last clue taped under teacher's desk)

Ciphertext for the clues:

First Clue: JUTZ ZNXUC GCGE G JAIQ (contained on lab file activity sheet)

Second Clue: ZDJLT KNQRWM CQN CRVN (teacher should write this on the hidden sheet)

Last Clue: GXFN GRZQ WR PB GHVN (teacher should write this on the hidden sheet)

The shift used for the first, second, and last ciphers are 6, 9, and 3 (respectively).

This activity is meant to reinforce students' understanding of Caesar Ciphers while highlighting the benefits of collaborating to solve decryption problems.

Start of Programming - 10 minutes or until the end of class
Students begin programming in class. The teacher looks for common problems encountered by students and encourages collaborative problem-solving without plagiarism.
Programming Assignment and Starter Code:
08.Programming

A lab file outlines the requirements for the programming assignment. Students begin work on the programming assignment with language-specific starter code. The starter code lab file guides student planning, and includes the following: a) a description of the assignment, b) important guidelines to simplify programming tasks, c) a description of difficulties students will likely encounter, d) programming best practice concepts, e) language-specific programming prerequisites, f) directions for establishing a test data set, and g) a Global Constants block.

The lab activity sheet, language-specific starter code, and complete language-specific programming solutions can be found in the Programming Folder. These resources are provided in the following languages: Python, Processing, and Pseudocode language description advanced by the College Board.

END OF DAY 1

DAY 1				DAY 2	
Teacher-led discussion on Caesar Cipher, Ciphers, Cryptography, and Programming	In-Class Encryption Activity	In-Class Internet Activity	Start Programming	Continue Programming	In-Class Decryption Activity
	Programming	Assignment:	Encryption	and	Decryption

DAY 2

Continue Programming - 20 minutes
Students will continue to work collaboratively to solve programming issues.

In-Class Decryption Activity - 20 minutes
06.CaesarCipher_InClass_DecryptionActivity.docx

07.CaesarCipher_InClass_DecryptionActivity_Solutions.docx

In this activity, students decrypt four individual sentences. Each sentence is the first sentence from a great work of literature: *A Tale of Two Cities*, *The Color Purple*, *A La Recherché Du Temps Perdu*, and *Pride and Prejudice*.

Students can organize themselves in any way they wish to decode the four encrypted sentences. Students are not allowed to use Internet-based tools to solve the problem. Students write their solutions in the spaces provided on the lab file. Lastly, students write a few sentences in the space provided on the lab file explaining their decryption process.

The activity reinforces current student understanding about cryptography in general, and Caesar Ciphers specifically. Modern decryption efforts are highly collaborative exercises: by allowing students to organize themselves in any manner they wish to complete the decryption task, hopefully students are encouraged to be mindful not only of the decryption activity, but mindful as well of the best way to organize, collaborate and work towards a solution.

ACKNOWLEDGEMENTS

Resources:

- Khan Academy on Caesar Ciphers
- Internet Hands-On Caesar Cipher Tool
- Wikipedia Article on Caesar Ciphers