THE ITERATIVE PROCESS

Dan Bonarigo, Riverside Brookfield High School, Riverside, IL

THE ITERATIVE PROCESS

This module provides exposure to iteration and password security.

OVERVIEW

Prerequisite Knowledge: Students should have been introduced to strings, Booleans, user input and variables and conditionals.

Length of Completion: The CCL is designed to take approximately 150-200 minutes.

Learning Setting: This lesson is intended for a face-to-face learning environment.

Lab Environment: This CCL is designed to be taught in a classroom that provides access to Internet-connected computers containing an Integrated Development Environment (IDE) consistent with class' chosen programming language.

Activity/Lab Tasks: Students will write a program or programs that implement the use of For and/or While loops and conditional statements involving variables, Booleans, user input and strings.

01.IterativeProcess_Overview.docx
02.IterativeProcess_Presentation.pptx
03.IterativeProcess_CodeSegmentsandOutput_Activity.docx
04.IterativeProcess_CodeSegmentsandOutput_ActivitySolutions.docx
05.IterativeProcess_CodeSegmentsandOutput_ActivityTwo.docx
06.IterativeProcess_CodeSegmentsandOutput_ActivityTwoSolutions.docx
07.IterativeProcess_PasswordProgramming_Activity.docx
08.IterativeProcess_PasswordProgramming_ActivitySolutions.docx

LEARNING OBJECTIVES AND AP CSP ALIGNMENT

LESSON LEARNING OBJECTIVES

Students will:

1) Develop a program to validate passwords, and more specifically,
2) Validate the length of the password,
3) Validate uppercase characters, and
4) Validate a numeric character.

ASSOCIATED AP CSP SUB LEARNING OBJECTIVES

AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 1: CREATIVE DEVELOPMENT

- 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.
  - CRD-2.B.2 A code segment refers to a collection of program statements that are part of a program.
  - CRD-2.B.3 A program needs to work for a variety of inputs and situations.
  - CRD-2.B.4 The behavior of a program is how a program functions during execution and is often described by how a user interacts with it.
- LO CRD-2.C Students will be able to identify inputs in a program.
  - CRD-2.C.1 Program input is data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text.
  - CRD-2.C.4 Inputs usually affect the output produced by a program.
  - CRD-2.C.6 Input can come from a user or other programs.
- LO CRD-2.D Students will be able to identify outputs in a program.
  - CRD-2.D.1Program output is any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text.

- o CRD-2.D.2 Program output is usually based on a program's input or prior state (e.g. internal values).
- LO CRD-2.J Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.
    - o CRD-2.J.1 In the development process, testing uses defined inputs to ensure that an algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs.
    - o CRD 2.J.2: Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data.

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

- LO AAP-1.A Represent a value with a variable.
    - o 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.
    - o 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.
    - o 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.
- LO APP-1.B Determine the value of a variable as a result of an assignment.
    - o AAP-1.B.1 The assignment operator allows a program to change the value represented by a variable.
    - o AAP-1.B.3 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-1.C Represent a list or string using a variable.
    - o AAP-1.C.4 A string is an ordered sequence of characters.
- LO AAP-2.E For relationships between two variables, expressions, or values:
    - ▪ Write expressions using relational operators

- Evaluate expressions that use relational operators
  - ○ AAP 2.E.1 A Boolean value is either true or false
- LO AAP-2.H Students will be able to write and evaluate 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.

LESSON DETAILS

Overview of Lessons: The teacher should use the presentation to review and introduce the content and then allow students to complete the activities.

Lesson 1:

- Presentation (Slides 1-22)
- Code Segments and Output Activity

Lesson 2:

- Presentation (Slides 23-29)
- Code Segments and Output Activity Two

Lesson 3:

- Presentation (Slides 30-34)
- Password Programming Activity

LESSON 1

Presentation
02.IterativeProcess_Presentation.pptx

Slides 4-13 are used to review booleans, strings, user input, variables and conditionals (teachers may choose to skip, and review slides as needed).

Slide 14-22 introduces the FOR loop: once these slides are completed, students should start the activity.

Code Segments and Output Activity
03.IterativeProcess_CodeSegmentsandOutput_Activity.docx

04.IterativeProcess_CodeSegmentsandOutput_ActivitySolutions.docx

 Students should complete Code Segments and Output Activity immediately following instruction on slide 22. During this activity students read and interpret the code segments and write the appropriate output for each code segment. Once the activity is completed the teacher should check students' answers and redirect misunderstanding.

LESSON 2

Presentation
02.IterativeProcess_Presentation.pptx
Teachers should continue with the presentation using slides 23-29 to introduce the While loop.

Code Segments and Output Activity Two

05.IterativeProcess_CodeSegmentsandOutput_ActivityTwo.docx
06.IterativeProcess_CodeSegmentsandOutput_ActivityTwoSolutions.docx

Students should complete Code Segments and Output Infinitive Loop Activity immediately following instruction of slide 29. During this activity students read and interpret the code segments and write the appropriate output for each code segment. Once the activity is completed the teacher should check students' answers and redirect misunderstanding.

LESSON 3

Presentation

02.IterativeProcess_Presentation.pptx
Teachers will conclude the lesson with the presentation of slides 30-32.

Password Programming Activity

07.IterativeProcess_PasswordProgramming_Activity.docx

08.IterativeProcess_PasswordProgramming_ActivitySolutions.docx

Students should complete the Password Programming Activity immediately following instruction on slide 32. Although sample programs are given, teachers should help each student use their creativity to arrive at a unique solution to the programming tasks. Example solutions are provided in the solutions document.

The teacher should guide a discussion on music sampling and the ethical implications while presenting slide 33.

ACKNOWLEDGEMENTS