

# Coding Fundamentals Series

*Bob Brown – Kennesaw State University*

*Bob.Brown@Kennesaw.edu*

This is a series of two sessions, each planned for a class period of about 50 minutes. This was developed with older students in mind, but will probably work for any middle or high school grade.

This series will help students understand the principles of procedural or object-oriented computer programming, and may make learning a particular programming language easier. The examples can be presented in whatever language the students are learning.

**Prerequisites:** Working familiarity with computing equipment, such as desktop computers, tablets, or smart phones; access to a procedural or object-oriented programming environment.

## **First Session: Coding and Programming; Data and Operations**

- Lecture, with slides, covering the following (15 minutes)
  - Where do apps come from?
  - “Coding” and “programming”
  - Procedural and non-procedural coding with examples.
  - The programming environment: editing, compilation or interpretation.
  - “Nouns:” variables and data types.
- Class discussion and exercise: which data type shall we use? (With handout)
- Lecture, with slides covering
  - “Verbs:” a programming language’s operators.
    - Arithmetic operators
    - Logical operators
    - Other operators
  - The importance of syntax
- Class discussion and exercise: A program to multiply two numbers. (To end of period)

**Outcomes:** Students who complete this session successfully will be able to:

- Briefly explain the difference between procedural and non-procedural computer coding.
- Distinguish between variables and operators.
- Select suitable data types for variables, given the characteristics of the data.
- Explain the importance of planning before coding.

## Second Session: Algorithms and Data Structures

- Lecture, with slides, covering the following (15 minutes)
  - Control structures
  - Expressing an algorithm: pseudocode and flowcharts
- Class exercise: Given a number  $n$ , develop an algorithm to print all counting numbers from 1 to  $n$ . Modify the algorithm to print only numbers that are multiples of 3.
- Lecture, with slides: Data structures
  - Arrays
  - Stacks and queues
  - Linked lists, trees
- Class exercise: Develop an algorithm to print a multiplication table for numbers up to  $12 \times 12$ .

**Outcomes:** Students who complete this session successfully will be able to:

- Name the three control structures used in programming.
- Name and briefly describe at least one type of data structure.
- Express an algorithm using pseudocode.