Computer Programming 1 — Chapter 1 Vocabulary

Essential Standard: 1.00 Understand ethics, security and the history of computer programming
Indicator 1.01 Understand the evolution of computers and computer programming languages
Indicator 1.02 Understand numbering systems
Indicator 1.03 Understand proper program documentation, code comments & use cases

1st generation computers	used punched cards to input and store data; used binary numbers & vacuum tubes
3rd generation computers	used Integrated Circuits (IC)
4th generation computers	an entire CPU (computer) on one IC chip; micro processors; Apple I, IBM PC
IC or "chips"	replaced transistors
Ada Byron Lovelace	often called the first programmer because she wrote a program based on the design of the Analytical Machine; Babbage's chief collaborator on the Analytical Machine
control bus	carries control signals
RAM/memory	random access memory, volitale
Hollerith's Tabulating Machine	used for US Census; Holes were made to represent information to be tabulated were punched in cards; Successful
Low level language	this code is written to specific hardware, and will only operate on the hardware it was written for and has almost no abstraction from the hardware; reads machine code and assembly language
computer peripheral devices	scanners, printers
Data Bus / Address Bus	transfers data between the CPU, memory and other hardware addresses that indicate where the data is located & where it should go
Magnetic tape	gave computers the ability to read (access) & write (store) data quickly & reliably
computer	an electronic machine that accepts data, processes it according to instructions, and provides the results as new data
Punch cards	replaced by magnetic tape & high-speed reel-to-reel tape machines
ALU	performs logic and arithmetic operations; makes comparisons; Arithmetic Logic Unit
CPU	processes data and controls the flow of data between the computer's other units
input devices	keyboard, mouse, CD/DVD, diskette drive, light pen
High Level languages	provides strong abstraction from the hardware allowing a program to be written in a language that can run on multiple types of computers (running the same operating system)
Storage	HDD, DVD, CD, Flash memory; non-volatile
ENIAC	(1943) Secret military project during WWII to calculate trajectory of artillery shells; Electronic Numerical Integration and Calculator
Analytical engine	(1833) Perform variety of calculations by following a set of instructions (or program) on punched cards
Mark 1	(1944) Mechanical telephone replay switches to store information and accepted data on punch cards; Highly sophisticated calculator – unreliable
Motherboard	contains expansion boards, clock rate, memory, SRSM, bus, data bus/address bus, control bus
Machine code	understood directly by the CPU, written in Hex code
Cache	high speed memory; L(Level) 1 cache is very high speed and stores instructions executed over and over; L2 cache is a slower and larger version of L1 cache
Apple 1	name of the first Apple computer
Wozniak & Jobs	introduced the first Apple computer
Static Random Access Memory	High-speed memory referred to as cache; Used to store frequently used data for quick retrieval

Computer Programming 1 — Chapter 2 Vocabulary

Essential Standard: 2.00 Understand the Solution Development Process
Indicator 2.01 Understand the Programming Process
Indicator 2.02 Understand Problem Solving Tools
Indicator 2.03 Understand proper program documentation, code comments & use cases

The Programming Process	A computer program is a list of instructions that contain data for a computer to follow; different programs are written with different languages
5- step programming process	Identify the Problem, Design the Solution, Write the Program, Test the Program, Document and Maintain the Program
Problem Solving Tools	Programs are created to solve problems; a solution must be designed prior to coding; one method of designing a solution to a problem is to create an algorithm.
Algorithms	A list of steps to solve a problem written in plain English; can be written out and numbered in the order in which they should be executed; should be written as extensive as necessary to outline the solution; it will not only tell your program what to do but how to do it.
Pseudocode	A mix of English language and code that represents what you want your program to do; helps you determine how you want the program to work as well as what variables and methods/functions you will want to include; helps you work through your logic, reducing the number of errors and potential re-writes you will have to do.
Flowchart	tool in programming process that use symbols and text to give a visual representation of a solution to a problem; helps the programmer begin to plan the programming project; provides a visual representation of the algorithm or process
Ovals (in a flowchart)	used at the beginning of the flowchart with the word “Start” and then “End” should be at the end of the flow chart or a process.
Parallelogram (in a flowchart)	used to show raw materials used for 'ingredients' & to show the finished product (Input/Output - Get/Display)
Rectangles (in a flowchart)	used to show processes/commands, eg. 'Bake Cake' (these are activities)
Diamonds (in a flowchart)	holds questions that resolve into True/False; used for decisions that divide into two options; also control loops
Structured Programming	A technique that has proven to be very effective in solving programs as well as in modifying solutions; the principles are fundamental to procedure-oriented programming and are also relevant to object-oriented programming; the ability to express a problem solution using only three basic patterns of logic
Simple Sequence Control Structure	a basic control structure that represents the computer's ability to execute instructions in a step-by-step, sequential manner. (example: directions to get to the school - steps must be followed sequentially)
Conditional Control Structure	a basic control structure that represents the computer's ability to make a decision. (example: provide alternate directions if there is a blocked intersection)
Iteration Control Structure	a basic control structure that represents the computer's ability to repeat a series of instructions within a loop or infinite loop; every loop must include a statement that defines how many times to execute the loop steps OR under what condition to continue OR when stop the looping process.
Loop	a series of repeated instructions (an **Infinite loop** would continue (keep repeating) without a way out)
Complexity	As a program become more complex documentation becomes more important (a program might be 100,000 lines of code)
Comments	documenting your programming code; can be on their own line or after a line of code; begin with a ' and turn the text green.
Use Cases	Define the interactions between the "actor" and the system; describes the sequence of steps between the actor (user) and the system necessary to complete a goal.
Actor	class of users, roles users play, or other systems, in other words, the actor is any user of the system.
Functional Requirements	behaviors that the software/system is intended to perform

Computer Programming 1 Chapter 3 Vocabulary

Essential Standard: 3.00 Apply Procedures to Construct Windows Forms
Indicator 3.01 Apply Controls associated with the Windows Form
Indicator 3.02 Apply Properties associated with Controls

Application A program which makes the computer a useful tool.

Assignment statement Uses the equal sign to give the object property on the left of the equal sign the value on the right of the equal sign.

Body The statements in a procedure.

Click event procedure A procedure that executes in response to a mouse Click event.

Code window The part of the IDE that displays the Form1 module window where program statements are entered.

Comment Text that explains and clarifies program code for other programmers; comments are preceded by a single quotation mark.

Compiler Converts a program to a language that the computer understands.

Control Used to create a control class object that the user can interact with.

Design window Displays the application interface and allows control class objects to be added, deleted, and sized.

Event Occurs when the user interacts with an object.

Event-driven application Executes code in response to events.

Event handler See Event procedure.

Event procedure Block of code that executes in response to an event.

Form A control class object that is an application interface. Contains a title bar, system menu, and Minimize, Maximize, and Close buttons.

IDE (Integrated Development Environment) Used to create or modify a Visual Basic .NET application.

Interface What appears on the screen when an application is running.

Menu bar The part of the IDE that contains the names of menus that contain commands. Can also be added to an application with a MainMenu control.

Numeric expression Formed with arithmetic operators.

Operator precedence The order in which operators are evaluated in a numeric expression.

Procedure A block of code written to perform specific tasks.

Program code A set of instructions in an application.

Project The set of files that make up a Visual Basic .NET application.

Project Explorer window The part of the IDE that lists the files in the current project.

Properties window The part of the IDE that lists the properties values of an object.

Property The part of a control object that defines its appearance, behavior, position, and other attributes.

Select Clicking an object, which displays handles.

Statement A line of code.

Solution Explorer window Used to switch between the Design and Code windows.

Toolbox The part of the IDE that contains controls that are used to add objects to a form.

Visual Basic .NET Object-oriented programming language used to create Windows, Web, and command-line applications.

Computer Programming 1 **Chapter 4 Vocabulary**

Essential Standard: 4.00 Understand Variables and Naming Conventions
Indicator 4.01Understand Variables and Data Types (5%)
Indicator 4.02 Understand Object Naming

Breakpoint A statement that has been marked as a stopping point.

Constant A named memory location which stores a value that cannot be changed from its initial assignment.

Debugging The process of getting an application to work correctly.

Declaration statement A statement used to create a variable or constant.

Coding Creating the interface and writing the program code.

Design How an application's interface will look and how the program code will be written.

Exception See Run-time error.

Floating point A data type that can represent values with numbers after the decimal point.

Function A procedure that performs a task and then returns a value.

Global declaration A declaration outside the procedures of a program. Also called module-level declaration.

Integer division Division performed with the \ operator to return only the whole portion of the quotient.

Keyword Identifier reserved by Visual Basic .NET.

Local declaration A declaration at the beginning of a procedure.

Logic error An error caused by syntactically correct statements that produce unexpected results. Also called semantic error.

Module-level declaration See Global declaration.

Modulus division Division performed with the **Mod** operator to return only the remainder portion of the division operation.

Prompt A label placed near a text box describing the expected input from the user.

Run-time error A syntax or logic error that halts a program at run time. Also called an exception.

Scope The set of statements that can be accessed by a declared variable or constant.

Semantic error See Logic error.

Specification Definition of what an application should do.

String A set of characters.

Syntax error An error caused by a statement that violates the rules of Visual Basic .NET.

Testing The process of running an application and entering data to test different possibilities to reveal any bugs.

Text box An object that allows the user to enter a value.

Variable A named memory location that stores a value.

Watch window The part of the IDE that can be used to examine values.

\ Arithmetic operator used to perform integer division.

Boolean A data type used to represent True or False.

Char A data type representing a single character.

Const Statement used to declare a constant.

Date A data type representing dates and times.

Decimal A data type representing currency.

Dim Statement used to declare a variable.

Double A data type representing very large positive or negative real numbers.

False One of two possible Boolean values.

Integer A data type representing positive or negative integers.

Long A data type representing very large positive or negative integers.

Mod Arithmetic operator used to perform modulus division.

Nothing Keyword that can be used in place of an empty string for clearing labels, and so on.

Short A data type representing positive or negative integers.

Single A data type representing positive or negative real numbers.

Step Into button Clicked to step through a program in break mode. Found on the Debug toolbar.

String A data type representing a string.

TextBox control Used to add a TextBox control class object to a form. Properties include Name, Text, and Alignment. Events include TextChanged.

True One of two possible Boolean values.

Val() A function that takes a string and returns a number corresponding to the numeric characters.

Computer Programming 1 **Chapter 5 Vocabulary**

Essential Standard: 5.00 Apply Programming and Conditional Logic
Indicator 5.01 Understand different types of programming errors.
Indicator 5.02 Understand Breakpoint, Watch Window, and Try And Catch to Find Errors.
Indicator 5.03 Apply Operators and Boolean expressions.
Indicator 5.04 Apply Decision-making Structures.
Indicator 5.05 Apply Looping Statements.

Accumulator A variable used to store a number that is incremented by a changing amount.

Concatenated Strings that have been joined.

Concatenation The process of joining two or more strings into one string.

Flag A condition used to signify that a loop should stop executing.

Infinite loop A loop that continues forever.

Input box A predefined dialog box that accepts input from the user.

Iteration Also called looping.

Loop A set of statements that repeatedly perform a task based on a condition.

Loop structure A statement that executes a loop as long as a condition is true.

Looping Repeating one or more statements.

Members Properties and methods of a class.

Method A procedure in a class.

Object A variable that is declared with a data type that is a class.

Pattern matching Allows wildcard characters (?, *, #), character lists, and character ranges ([A–M, Z]) to match strings using the Like operator.

Robust Describes an application that performs even under unexpected circumstances.

Shared method A method that is used with a class, not with a particular object of that class.

Sentinel See Flag.

Structure A simple form of a class.

Substring A portion of a string.

Textual comparison Comparing characters without distinguishing case.

Unicode A digital code that represents every letter of an alphabet and symbols of every culture. Each code uses 16 bits or 2 bytes.

***=** Assignment operator that multiplies the value on the right of the statement and the current value of the variable on the left and then updates the variable to store the new value.

/= Assignment operator that uses the value on the right of the statement to divide the current value of the variable on the left and then updates the variable to store the new value.

\= Assignment operator that uses the value on the right of the statement to divide the current value of the variable on the left and then updates the variable to store the integer portion of the new value.

^= Assignment operator that raises the current value of the variable on the left to the power of the value on the right of the assignment statement and then updates the variable to store the new value.

&= Assignment operator that concatenates the string on the right of the statement and the current string of the variable on the left and then updates the variable to store the new string.

& Used to concatenate two or more strings.

AscW() Function that returns the integer Unicode value that corresponds to a character argument.

Char **structure** Used to manipulate characters. Methods include ToLower() and ToUpper().

ChrW() Function that returns the character corresponding to an integer representing a Unicode value.

Do...Loop Statement that repeatedly executes a loop as long as a condition is True.

For...Next Statement that executes a loop a fixed number of times.

InputBox() Function used to generate a predefined dialog box that has a prompt, a text box, and OK and Cancel buttons.

Like Operator used to perform textual comparison on strings and pattern matching using characters such as ?, *, #, [].

Space() Function used to generate a string of spaces.

Step Keyword used in a For...Next statement to increment or decrement the counter by a set amount.

String class Used to manipulate strings. Properties include Chars and Length. Methods include Concat(), Compare(), IndexOf(), Insert(), PadLeft(), PadRight(), Remove(), Replace(), Substring(), Trim(), ToUpper(), ToLower(), TrimEnd(), and TrimStart().

vbCrLf Built-in constant that represents a carriage return-linefeed combination.

vbTab Built-in constant that places a string in the next field of eight characters.

Computer Programming 1 **Chapter 6 Vocabulary**

Essential Standard: 6.00 Apply tools to obtain and validate user input.
Indicator 6.01 Apply Procedures to Develop Menus, ListBox, and ComboBox objects.
Indicator 6.02 Apply Procedures to Develop Message, Input, and Dialog Boxes.
Indicator 6.03 Apply Procedures for Validation of User Input.

ComboBox control Displays a textbox for typing an entry and also an arrow that can be clicked to choose an item from a list; a scrollbar appears in the list if necessary.

ComboBox control properties Name, Items, Text, Enabled, Sorted, Visible

Component tray An icon placed in the lower gray area when you add a MenuStrip to your form

Concatenation The process of joining two or more strings into one string.

Convert Class Methods Will throw a runtime error if the string is not numeric

Dialog box Used to get information from the user.

InputBox function Function allowing the user to input information; must display a prompt in a dialog box, wait for the user to either hit the enter key or click a button, then returns the information from the textbox as a string.

ListBox control Displays a list of items for the user to select from; a scrollbar appears if the list is longer than the list box.

ListBox control properties Name, Items, Text, Enabled, Sorted, SelectedItem, SelectedIndex, Visible

MenuStrip control Adds a menu bar to your form; prefix is mnu

MessageBox control A predefined dialog box that displays information to the user.

Method A procedure in a class.

SelectedIndex Property A ListBox control property that allows the programmer to compare what the user has selected in the list box with a value.

SelectedItem Property A ListBox control property that allows the programmer to compare what the user has selected in the list box with the item itself as a string value.

Title Parameter Displayed in the title bar at the top of the dialog box

TryParse Used in an assignment statement or in an if statement as it returns true/false; if true, it will convert and load the number into the numeric variable, but if false, it will load 0 into the numeric variable.

TryParse(String, Int16) Converts the string representation of a number to its 16-bit signed integer equivalent. A Boolean return value indicates whether the conversion succeeded/failed (true/false).

TryParse(String, Int32) Converts the string representation of a number to its 32-bit signed integer equivalent. A Boolean return value indicates whether the conversion succeeded/failed (true/false).

TryParse(String, Double) Converts the string representation of a number to its double equivalent. A Boolean return value indicates whether the conversion succeeded/failed (true/false)

Chapter 6 Visual Basic

formName.Hide() hide the dialog box

formName.Show() show the dialog box

Items.Add() Function used to add an item to a list at run time

Items.Clear() Function used to delete the contents of the list box

Items.Remove() Function used to delete a specified item from the list

Computer Programming 1 **Chapter 7 Vocabulary**

Essential Standard: 8.00 Apply Advanced Logic
Indicator 7.01 Apply sub procedures/methods and user defined functions
Indicator 7.02 Apply One-Dimensional Arrays
Indicator 7.03 Apply Built-in Math Class Functions
Indicator 7.04 Apply Built-in String Functions

Arrays a "container" that holds more than one value and each value is called an element

Class a string data type that includes properties and methods called members.

Compare() function compares two specified String objects and returns an integer that indicates their relative position in the sort order

Concat() function Creates the string representation of a specified object, usually two or more strings

Element each value in an array; each element of the array must be the same data type

Equals() function Determines whether this instance and another specified String object have the same value

Event Procedure code that responds to an event that executes in response to the user; an example is a button click

Index position points to the location of a value in an array

IndexOf() function Reports the index of the first occurrence of the specified Unicode character (or string)

Insert() function Inserts a specified instance of String at a specified index position in this instance

Length a property that will give the length (number of elements) of any declared array

Math functions works with either a decimal or a double; absolute, Sign, Square Root, Round

Modularization the breaking up of a program

Postcondition a statement of what must be true at the end of the execution of a procedure if the procedure has worked properly

Precondition assumptions or initial requirements of a procedure; you should include this above your procedure

Reference Parameter does not create a new place in memory (storage); represents the same storage location as the variable given in the argument "call"

Remove() function Deletes all the characters from this string beginning at a specified position and continuing through the last position

Replace() function Returns a new string in which all occurrences of a specified string in the current instance are replaced with another specified string

Sub Procedure a block of code that will execute in response to a "call" from inside another block of code; when the Sub is called, its code statements are executed

ToLower() function Returns a copy of this string converted to lowercase

ToUpper() function Returns a copy of this string converted to uppercase

Trim() function Removes all leading and trailing white-space characters from the current String object

TrimEnd() function Removes all trailing occurrences of a set of characters specified in an array from the current String object

TrimStart() function Removes all leading occurrences of a set of characters specified in an array from the current String object

Computer Programming 1 **Chapter 8 Vocabulary**

Essential Standard: 8.00 Apply Procedures to Develop Graphic Applications
Indicator 8.01 Understand coordinate systems.
Indicator 8.02 Apply Procedures to Create Picture Boxes using Images.
Indicator 8.03 Apply Animation and Graphic Methods in a Windows Form

AutoSize SizeMode within the PictureBox Control; resizes the PictureBox to fit the image

CenterImage SizeMode within the PictureBox Control; puts the image in the middle of the PictureBox

Default Coordinate System made up of rows and columns of pixels

Drawing surface defined by assigning an object's surface to a Graphics object; a grid consisting of a set of points with x & y values where each point is a pixel

Enabled (default) True if the timer can respond to user interaction, otherwise false

Graphics class can be used to draw shapes; class contains methods for creating circles, lines, rectangles and other shapes on a drawing surface.

Image sets the image to display in the PictureBox

Interval sets the time in milliseconds between elapsed events

PictureBox control displays an image that can be one of the following formats: Bitmap, GIF, JPEG; name of a PictureBox should start with pic

Point Structure A point has an x-coordinate and a y-coordinate that together indicates a specific location; you can declare a point with the X and Y values included when the keyword New is used

Normal SizeMode within the PictureBox Control (this is the default setting), places in the top-left corner of the Picture Box

Origin the top left corner of your form, 0,0; the bottom right corner is the width and height of the control

Pixels are the smallest points you can locate on the form

SizeMode Controls the image sizing and position (how the image is displayed): Normal (default), StretchImage, AutoSize, CenterImage and Zoom

Size Property (of an object) stores both a height & width and can be used to determine the point in the lower-right-hand of an object

Start starts by raising the Elapsed event by setting Enabled to True

Stop stops by raising the Elapsed event by setting Enabled to False

StretchImage SizeMode within the PictureBox Control; resizes the image to fit the PictureBox

Timer Control displays in your component tray and generates recurring events; name of a timer should start with tmr

Zoom SizeMode within the PictureBox Control; resizes the image to fit the PictureBox but maintains the original ratio