“Visual Basic .Net” Online Course

Glossary

**Abstraction** – Hiding internal details of an object from the user.

**Algorithm** – The problem solving series of steps or method taken for converting inputs of a specific problem into meaningful outputs.

**Architecture** - The design of the application including the break down of parts.

**Argument** – A means by which you communicate data to a procedure by passing the argument and its data type.

**Arithmetic-logic unit (ALU)** – A computer component that performs the primary logical and math operations of the computer.

**Artifact** Data, source code, and documentation created or used during the development process.

**Assembler** – Translates source code into machine language.

**Assembly language** – A computer language designed for a specific machine and specific microprocessor. Mnemonics are used in place of 0’s and 1’s to aid in the programming process. The mnemonic corresponds to a machine operation (load, sum) and assigns addresses and storage locations.

**Attribute** - Characteristics describing an object referred to as properties in Visual Basic.

**BASIC (Beginners All-purpose Symbolic Instruction Code)** – General purpose programming language developed by Dartmouth College in the 1960s to teach students how to use computers.

**Beta release** A preliminary version of a software application provided to a predetermined audience to detect errors and provide feedback to the developers or the software.

**Boolean** – An expression that can be evaluated to a True or False condition.

**Build (Debug)** - Debug build contains debugging information and creates an executable file that can run on any system that contains Microsoft Windows and the support files for the application.

**Build (Release)** – Release build is optimized to run efficiently and doesn’t contain the debugging information in the creation of the executable file.
Class – A category of objects that is a pattern or blueprint for creating an object.

Compiler – Used to translate source code into machine code called object code.

Computer – A device containing a central processing unit (CPU), memory and other electronic devices used to translate instructions and data into information.

Connector – Circle shaped symbol use in flowcharts indicating where flowlines are connected. They are used in pairs showing flowline in and flowline out.

Constant – A value that cannot be changed during program execution.

Control - An object used on a graphical interface (GUI), such as a command button, option button, label or text box.

Central Processing Unit (CPU) - The component within the computer where manipulation of data takes place and controls the other parts of the computer system.

Class – A template dictating how an object created from the class appears and functions.

Data flow diagram – Diagram illustrating inputs, outputs and internal processes as data flows through the application.

Data Input or Output - Parallelogram shaped symbol use in flowcharts indicating the input data and results (output) to be displayed.

Decision – A diamond shaped symbol use in flowcharts indicating a decision structure. Diamond symbols always have two (2) flowlines out. One flowline is labeled the “yes” or “true” branch and the other is labeled the “no” or “false” branch.

Declaration statements – Used to define names of data and procedures.

Dynamic help – Displays a list of help topics as a tool window that will change as you move your cursor from one window to another in the process of completing different tasks.

Encapsulation – The combining of an object’s attributes and behaviors into one package called a class.

Executable statements – Used to perform a specific actions.

External documentation – Written manual detailing the program’s operation to allow another person to modify the code if it becomes necessary. The manual may contain flowcharts, pseudocode, design requirements, input and output requirements and other supplementary documentation to aid in the maintenance of the program.
**Event** – Action taken by the user of the application, such as a click, pressing a key, dragging or scrolling and recognized by the computer program.

**Event-driven programming** - Key concept used by Visual Basic whereby the application is designed to respond to actions taken by the user.

**Event procedure** – A block of code written to respond to an action taken by the user.

**First generation electronic computers** – Computers using vacuum tubes during the period from 1946 to 1956. Difficult to program and used only machine language.

**First generation computer language** – Used 0’s and 1’s meaning either an on or off state or positive charge of negative charge understood by the computer. The language was computer dependent and called machine language.

**Flowcharts** – Specific shaped diagrams representing different processes used by the software to show flow of the program through the application.

**Flowline** – Connect the flowchart symbols and show the process flow during the execution of the program.

**Form** – A window where you can place elements such as buttons and text boxes.

**Fourth generation computer language (4GL)** – Used nonprocedural language to develop software applications with little or no technical assistance. Nonprocedural languages specify what has to be accomplished rather than to do it.

**Fourth generation electronic computers** – Computers using microprocessors during the period from 1980 to present day.

**Functional requirements** – Requirements expressing a function that an application must perform.

**Graphical user interface (GUI)** - A graphical interface allowing the user to interact with the application containing icons, buttons and menu bars.

**Handle** – A small square on a design control such as a label box allowing resizing.

**High-level language** – A programming language using sentence-like statements that will generate multiple statements when it is translated into a machine language.

**Identifier** – A software engineer supplied name for a constant, control, procedure or variable.

**Image** – A control containing a image or picture having a property for resizing.
Inheritance – The ability of newly created class to receive the objects, properties, methods and other attributes of another class.

Interactive development environment (IDE) – A software application environment allowing software engineers to create, edit, compile and execute code.

Interface – A common phase used by software engineers describing the boundary where the software application interacts with the user or exchanges messages with other objects.

Internal documentation – Comments made inside the code to explain the processes of the program.

Interpreter – Translates each source code statement one at a time into machine code before executing it.

Iteration – Passing through the body of a loop.

Joint application development (JAD) – Process used to accelerate the requirement analysis by having end users and software engineers working together in intensive interactive design sessions.

Label – A control displaying text as a caption and cannot be altered by the user.

Legacy application – An application that has been in production for a period of time.

Loop – One of the control structures providing for the repetition of a statement.

Machine language – A computer language composed of 0’s and 1’s (binary digits) and used directly by a computer without translation. The binary digits represented computer instructions and information.

Mainframe – A large capacity computer that can cost millions of dollars with massive amounts of memory and rapid processing power used for large business, military and scientific applications.

Maintenance – An ongoing process associated with modifying code to repair or enhance an application’s functionality.

Methods – Operational code that can process or manipulate data.

Microcomputer – Most commonly purchased by the general public and usually referred to as a personal computer (PC).

Microsoft Development Environment (MDE) – It is the Visual Studio . Net programmer interface that is used to create programs for the end user.
Minicomputer – A mid-range computer that is not as fast as a mainframe and has smaller storage capacities normally used by offices, universities, and research laboratories.

Object – A unit of computer information containing data as well as procedures.

Object-oriented programming (OOP) – Using an object-oriented language to create a program containing one or more objects.

Output window – Displays status messages such as build errors at run time.

Paradigm – A way of thinking such as using the object-oriented paradigm used in Visual Basic.

Peripherals – Hardware attached to the computer that include devices such as a monitor, keyboard, printer and other components.

Pixel – A basic picture element making up an image displayed on a computer monitor representing the smallest point that can be located or addressed.

Polymorphism – Multiple forms having the same name procedure name but performing different tasks.

Predefined Process – A rectangle shaped symbol with vertical lines next to each side used in flowcharts indicating a group of previously defined statements.

Primary storage – A component of the computer that temporarily stores program instructions and data being used by the instructions.

Procedure – A unit or block of code detailing instructions to be taken to accomplish a task.

Procedure languages – A classification of computer languages that provide instructions on what to do and how to do it and are generally referred to as high-level languages.

Problem solving – The process of using a set of inputs, acting on the inputs by doing some calculation with them and producing a result.

Process – The order in which the development activities or “software process” is performed.

Processing – Rectangle shaped symbol use in flowcharts indicating an operation performed by the computer program such as calculation and assignment of values.
Properties Window – A control feature of the MDE that allows the programmer to manage the appearance of each object that has been placed on the form and the appearance of the form.

Prototype – A preliminary application or mockup that demonstrates features of the software project under construction often used in requirement analysis to facilitate the design process.

Pseudocode – English like phrases used to represent the required steps to be taken in program code referred to as an algorithm.

Random access memory (RAM) – Volatile memory used for short term storage and is lost when the computer is turned off. It is internal temporary memory that stores data and instructions during processing.

Requirement analysis - The process of obtaining a complete written statement of the software’s required appearance (GUI), behaviors, functionality and performance.

Rapid application development (RAD) – The process of creating workable computer systems in a very short period of time. Visual Basic is considered a RAD development tool since it uses visual programming and other tools for building a GUI with functionally quickly.

Read only memory (ROM) – Non-volatile memory that can only be read from and cannot be written to. ROM chips are usually created by the manufacturer with dedicated programs already burned in or stored. The program is not lost when the power is turned off on the computer.

Second generation computer language – Used a low-level language consisting of mnemonic which are abbreviations for program instructions. The language used is called assembly language and is machine-dependent.

Second generation electronic computers - Computers using transistors during the period from 1957 to 1963.

Secondary Storage – A device such as disks and tapes where information is stored prior to being transferred into primary memory for processing.

Solution Explorer – Its function is to group the elements needed to build and run a solution.

Source code – Program instructions that need to be translated into the machine language before it is executed by the computer.

Supercomputer – An extremely powerful computer used for tasks requiring complex calculations with hundreds of thousands of variables.
**Statement** - A line of code used in programming Visual Basic expressing a complete thought or action.

**Syntax** – Rules that must be followed in the process of creating programming statements.

**Terminator** – Oval shaped symbol use in flowcharts indicating the beginning or ending points of the computer program. The terminator can have only have one flowline, either in or out.

**Third generation computer language (3GL)** – Used procedure languages with English like statements (Print, Display) to create programs without having to understand specific details on how the computer processed the data.

**Third generation electronic computers** – Computers using integrated circuits during the period from 1964 to 1979.

**Toolbox** – A portion of the developers interface containing all of the visual tool or controls used to create what appears on the screen for the user of the program.

**Validation** – Process of ensuring a software application performs according to functional requirements.

**Variable** - A memory location that is referred to by name that can be changed during the application’s execution.

**Verification** – Process of ensuring a software application is being built according to the plan.

**Windows Forms Designer** – Provides the means for the programmer to create the interface of the form for the user.

**Workstation** – Used in engineering and other application requiring powerful graphics and mathematical capabilities.