NAME:Najeebat ahmed opeyemi

MATRIC NO:16/scio3/001

DEPARTMENT:computer science

COURSE:csc201

 HISTORY OF VISUAL BASIC TILL DATE

Visual Basic 1.0 was introduced in 1991. The drag and drop design for creating the user interface is derived from a prototype form generator developed by [Alan Cooper](https://en.wikipedia.org/wiki/Alan_Cooper) and his company called *Tripod*.Microsoft contracted with Cooper and his associates to develop Tripod into a programmable form system for [Windows 3.0](https://en.wikipedia.org/wiki/Windows_3.0), under the code name *Ruby* (no relation to the later [Ruby programming language](https://en.wikipedia.org/wiki/Ruby_%28programming_language%29)). Tripod did not include a programming language at all. Microsoft decided to combine Ruby with the Basic language to create Visual Basic. The Ruby interface generator provided the "visual" part of Visual Basic and this was combined with the "EB" Embedded BASIC engine designed for Microsoft's abandoned "Omega" database system. Ruby also provided the ability to load [dynamic link libraries](https://en.wikipedia.org/wiki/Dynamic-link_library) containing additional controls (then called "gizmos"), which later became the [VBX](https://en.wikipedia.org/wiki/Visual_Basic_Extension) interface

Visual Basic is Microsoft's high-level object-oriented [rapid application development](https://en.wikipedia.org/wiki/rapid_application_development) environment for the Windows platform. The first versions of Visual Basic were intended to target Windows 3.0 (a version for DOS existed as well), however it was not until version 3.0 for Windows 3.1 that this programming language gained large-scale acceptance in the shareware and corporate programming community.

Using drawing tools that resemble those found in hardcopy page layout programs or PhotoShop, VB programmers make user interfaces by drawing controls and other UI components onto forms. The programmer then adds code to respond to user interactions with the controls (for example, clicks, drag and drop, etc) known as events. The code can trigger events in other controls (for example, by displaying text or an image), execute procedures (run some algorithm based on the values entered in some control, output data, do business logic, etc), or almost anything else one might do in code.

Visual Basic can be considered to be an interpreted language like its [Basic](https://en.wikibooks.org/wiki/Subject%3ABASIC_programming_language) ancestor, with appropriate modifications to accommodate [object-oriented programming](https://en.wikibooks.org/wiki/Object_Oriented_Programming), and has implicit type conversion. That is, the VB development environment goes to great lengths to format (and aid the user in formatting) programming code so that it conforms to executable syntax. For example, VB will appropriately change the case of newly typed variable names to match those that have been declared previously (if they have been declared at all!). Traditionally, VB is known for compiling programs into pseudo-code (p-code, similar to Java's byte code) which is interpreted at runtime, requiring the use of dynamically-linked libraries (for example, VBRUN300.DLL for version 3 of Visual Basic, circa 1992) but newer versions can compile code into something more closely resembling the efficient machine code generated by C-like compilers. VB6 can be compile either into p-code or into native code; in fact VB6 uses the Microsoft C++ compiler to generate the executable.

For new Windows programmers, VB offers the advantage of being able to access much of the Windows UI functionality without knowing much about how it works by hiding the technical details. Although accessing low-level Windows UI functionality is possible, doing so in VB is as, or more difficult compared to such access using Visual C++ or other lower level programming languages. Recently [VB.NET](https://en.wikibooks.org/wiki/Visual_Basic_.NET) has gone a long way to fixing some of the limitations.

Using custom controls provided by Microsoft or third parties, almost any functionality that is possible in Windows can be added to a VB program by drawing a custom control onto a form in the project.

Visual Basic traditionally comes in at least entry level and professional versions, with various designations depending on Microsoft's contemporary marketing strategy. The different versions are generally differentiated by the number of custom controls included, and the capabilities of the compiler. Higher priced packages include more functionality.