

PROGRAMMING IN VISUAL BASIC HONORS

Course Description

Visual Basic for Windows in the Microsoft Visual Studio.Net introduces students to the major elements of the Visual Basic programming language. The course curriculum introduces the integrated development environment and building an application in that environment. Students will work with variables, constants, data types, and expressions. Problem solving and decision making are integrated as students also explore looping and multiple forms, using menus, common dialogs, procedures, functions and arrays, debugging, creating executable files, and distributing a Windows application. Concepts and methods of object-oriented programming and design, creating applications using a development cycle approach, and disciplined coding style are included.

Recommended for: Students interested in programming and computer science

Course length: Half year

Credits: .5

Level: II

Goals

- Develop understanding of and use the Visual Basic programming language as a solution tool for application and business problems
- Learn the process of programming using a development cycle approach
- Recognize that programming languages have common elements
- Seek and evaluate multiple solutions to a problem
- Reinforce the use of a problem-solving approach as a life skill
- Understand the interactions of computer hardware and software
- Distinguish between procedural and object-oriented programming
- Promote an atmosphere conducive to creativity and individuality
- Provide opportunities for class discussions of problem solutions with peers
- Encourage good ethical practices
- Compare the efficiency of program solutions

- Evaluate the functionality and usability of program solutions
- Cultivate a goal-oriented learning environment

Competencies

- Use the Microsoft.Net environment
- Use the Visual Basic programming interface
- Create a problem-solving solution with an algorithm
- Learn the phases of the development cycle to create applications
- Understand object-oriented programming (OOP) and design (OOD) and rapid application development (RAD)
- Describe the components of the Visual Basic integrated development environment and customization features
- Develop projects encompassing code and files to create and run a program
- Produce, maintain, and execute new applications
- Design, implement, and test a Windows application
- Understand how to write Visual Basic code
- Use tools and constructs including controls, data types, and code structures to develop Windows applications
- Distinguish among different types of variables and constants
- Learn to use If...Then...Else structures including looping
- Use strings in code and learn that strings can be manipulated
- Perform a test using logical and rational operators and access feedback
- Learn about coding procedures to write “For...Next,” “For Each...Next,” “Do Until,” and “Do While” statements
- Gain knowledge of manipulation and utilization of one- and two-dimensional arrays

- Find and correct syntax errors using debugging tools
- Investigate methods of evaluating variables and expressions

Course Content

I. Introduction to Visual Basic Programming Environment

- A. Visual Basic.Net and the Visual Basic programming language
- B. Program development
- C. Phases of the development cycle
- D. Object-oriented programming and design
- E. Rapid application development
- F. Algorithms
- G. The programming interface

II. Introduction to the Integrated Development Environment

- A. Basic components of Visual Basic integrated development environment
- B. Customize the environment
- C. Set properties on controls and navigate the code window
- D. Modify code in a project
- E. Run a project
- F. Use Visual Basic.Net Help features

III. Build an Application

- A. Design an application
- B. Work with forms
- C. Use controls—label, textbox, numeric up down, and button
- D. Change property values of controls
- E. Write and document code
- F. Evaluate and modify the program

IV. Use Variables, Constants, Data Types, and Expressions

- A. Declare variables and constants
- B. Use variables and constants within code
- C. Describe data types
- D. Convert between data types
- E. Code a form load event procedure
- F. Use the option strict statement
- G. Arithmetic expressions
- H. Order of operator precedence in code
- I. Pmt and format\$ functions

V. Modify Applications

- A. ComboBox control
- B. If...Then...Else... structure and nested structures
- C. Selection structure
- D. Validation of input
- E. Strings in code
- F. Rational and logical operators in code

VI. Looping and Multiple Forms

- A. Add forms to a project
- B. Change default of forms
- C. Display item lists on forms
- D. Use form controls
- E. Work with collections in code
- F. Code Do Until, Do While loops
- G. Code For...Next and For Each...Next loops

VII. Use Menus, Dialogs, Procedures, Functions, and Arrays

- A. Add menus to applications and shortcuts
- B. Common dialog boxes in an application for user interaction
- C. One- and multidimensional arrays in code
- D. Function procedure to return a value
- E. Subprocedures

VIII. Debugging, Executable Files, and Distribution

- A. Differences among syntax, logic, and run-time errors
- B. Locate errors during design time
- C. Locate errors during run time
- D. Set breakpoints
- E. Stepping to execute one statement at a time
- F. Evaluate variables and expressions
- G. Change values of variables using debugging tools
- H. Examine variables and objects
- I. Create executable files
- J. Create a setup program to distribute a Windows application

Instructional Methodologies

- Microsoft Visual Studio.net/Visual Basic Programming Language
- Software Help auxiliaries
- Internet Web site tutorials

- Individual programming exercises, assignments and projects
- Teacher demonstrations and guidance
- Peer learning
- Problem-solving approach
- Application programming
- Database management
- Peer assessment

Instructional Resources (Students)

- Microsoft Visual Studio.net, 2003, 7.1/Visual Basic Programming Language
- Teacher-prepared supplementary handouts
- Teacher-prepared exercises, assignments, projects
- Reference manuals
- Software Help auxiliaries
- Internet Web site tutorials
- Simulations

Teacher Resources

Texts—*Visual Basic.Net*, Second Edition, Diane Zak, Thomson Course Technology
—*Microsoft Visual Basic.Net*, Shelly, Cashman, Quasney, Thomson Course Technology, 2003
—*Microsoft Visual Basic.Net Reloaded*, Diane Zak, Thomson Course Technology, 2004

- Assorted Visual Basic, computer science, and programming reference texts—BITC Dept. Library
- Visual Basic Internet Web site Tutorials
- Visual Basic Software Help auxiliaries
- Teacher resources for *Microsoft Visual Basic.Net*—Instructor’s Manual, ExamView, Lecture Success System, PowerPoint Presentations, Laboratory Solutions, Chapter Reinforcements, Student Data Files, Interactive Labs, Help features
- Computer Science Teachers Association (csta) Repository for Visual Basic—Web site, Help, Podcasts

Curriculum Map

Terms I/III

- Introduction to the Visual Basic Programming Environment
- Introduction to the Integrated Development Environment
- Build Applications
- Variables, Constants, Data Types, and Expressions
- Modify Applications

- Application problems, programs, and projects

Terms II/IV

- Looping and Multiple Forms
- Menus, Dialogs, Procedures, Functions, and Arrays
- Debugging, Executable Files, and Distribution
- Application problems, programs, and projects

Technology Resources

- Computer laboratory classroom with Internet access
- Computer overhead projection system for demonstration/presentations
- Network printer
- Microsoft Visual Studio.net, 2003, 7.1/Visual Basic Programming Language

Methods of Assessment

- Algorithms
- Programs and projects
- Tests and quizzes
- Vocabulary and terminology
- Creativity
- Participation/work ethic
- Rubrics
- Class discussion
- Enrichment activities/extra credit assignments
- Final examination

Academic Support—Enrichment Topics and Activities

- Independent programming projects of increasing levels of difficulty
- Peer evaluation of student programs
- Troubleshooting/debugging of peer programs
- Troubleshooting/debugging of sample programs
- Research and document the similarities and differences between Basic Programming and Visual Basic Programming
- Research similarities among Visual Basic and other popular programming languages

Global Awareness/Global Economy

- Recognize that the problem-solving skills used in this class are transferable to life problems and their solutions in a global economy
- Understand that the code is used all over the world to solve problems
- The use of Microsoft Visual Studio.Net is used worldwide to create business, education, and Internet applications
- Awareness that program code is the access to global data and communications
- Need for continual upgrading and expansion of computer skills as a requirement for successful employment in a global economy
- Concept of “globalized professionals” (possessing qualifications for success in a global economy) and connection to computer careers and the insourcing and outsourcing of computer skills
- Technology creates greater interdependence among workers, organizations, and nations

Massachusetts Technology Literacy Standards and Expectations Reflected in the Course Competencies (2008)

Basic Operations

- G9-12: 1.2 Use online help and other support to learn about features of hardware and software, as well as to assess and resolve problems
- G9-12: 1.4 Explain effective backup and recovery strategies
- G9-12: 1.5 Explain criteria for evaluating hardware and software appropriate for a given task (e.g., features, versions, capacity)
- G9-12: 1.6 Demonstrate keyboarding techniques, including the use of keyboard shortcuts, to complete assignments efficiently and accurately.
- G9-12: 1.7 Identify and assess capabilities and limitations of emerging technologies

Internet, Networking, and Online Communication

- G9-12: 1.28 Explain and demonstrate effective search strategies for locating and retrieving electronic information

Ethics

- G9-12: 2.1 Demonstrate compliance with the school’s Acceptable Use Policy
- G9-12: 2.2 Explain issues related to the responsible use of technology (e.g., privacy, security)
- G9-12: 2.3 Explain laws restricting the use of copyrighted materials
- G9-12: 2.4 Identify examples of plagiarism, and discuss the possible consequences of plagiarizing the work of others

Society

- G9-12: 2.8 Design and implement a personal learning plan that includes the use of technology to support lifelong learning skills

Health and Safety

- G9-12: 2.15 Explain ways individuals can protect their technology systems and information from unethical users

Research

- G9-12: 3.2 Compare, evaluate, and select appropriate electronic resources to locate specific information
- G9-12: 3.4 Search for information within an electronic source (e.g., using the find command)

Problem Solving

- G9-12: 3.5 Explain and demonstrate how specialized technology tools can be used for problem solving, decision making, and creativity in all subject areas (2008 Introduction to Massachusetts Technology Literacy Standards for Grades 9-12—**Students should have the opportunity to learn how to write code in a commonly used programming language.**)

Communication

- G9-12: 3.10 Complete at least one online credit or non-credit course or tutorial; discuss the benefits and disadvantages of this type of learning

School-to-Career Skills Reflected in the Course Competencies

(*Reflect SCANS Skills—The Secretary’s Commission on Achieving Necessary Skills, U.S. Dept. of Labor)

- Development of foundation competencies
 - Basic Skills in reading, writing, listening, arithmetic, and mathematics*
 - Thinking Skills—thinking creatively, making decisions, solving problems, seeing things in the mind’s eye, knowing how to learn, and reasoning*
 - Personal Qualities—develop individual responsibility, self-management, and self-esteem*
- Development of computer literacy skills for lifelong learning
 - Information—acquiring and evaluating data, organizing and maintaining files, interpreting and communicating, and using computers to process information*
 - Technology—selecting equipment and tools, applying technology to specific tasks, and maintaining and troubleshooting technologies*
 - Systems—understanding technological systems*
- Identification of transferable workplace skills
 - personal and career development skills
 - research and investigation skills*
 - decision making skills*
 - creative thinking skills*
 - critical thinking skills
 - time management skills*
 - problem-solving skills*
 - organizing skills*
 - transfer learning strategies*
 - apply information to problems and tasks*
 - analyze/interpret information and tasks*
 - ethics skills*
 - teach others new skills*

- communicating effectively*
- visioning an end result and using technology to produce an outcome
- Preparation for internships, education after high school and employment

AHS School-Wide Academic Expectations Reflected in the Course Competencies

- Students will demonstrate satisfactory achievement in the standards-based curriculum at Agawam High School
- Students will communicate effectively through listening
- Students will communicate effectively through reading
- Students will communicate effectively through speaking
- Students will communicate effectively through writing
- Students will use scientific and mathematical processes to interpret and evaluate information and solve problems across the curriculum
- Students will develop proficiency in information and communication technology literacy skills and will use appropriate tools to identify and solve problems across the curriculum
- Students will participate in activities to foster individual interests and fulfill individual potential
- Students will become knowledgeable of the changing employment market and acquire skills for career exploration

Massachusetts Common Core of Learning Reflected in the Course Competencies

- Read, write, and communicate effectively
- Use mathematics, art, computers, and other technologies effectively
- Define, analyze, and solve complex problems
- Acquire, integrate, and apply essential knowledge
- Study and work effectively
- Demonstrate personal, social, and civic responsibility

Computer Science Teachers Association (csta) Model Curriculum for K-12 Computer Science Reflected in the Course Competencies

Level III Objectives

Topic 1: Program Design and Problem Solving

- Given a problem statement, write a natural language procedure for solving the problem
- Use design tools (such as flowcharts or UML diagrams) to express a solution to a problem
- Solve a problem by applying a problem-solving process and translate the solution into a program that uses appropriate data and control structures
- Illustrate the “is-a” and “has-a” object-oriented concepts
- Test a solution to ensure that it meets its stated requirements

Topic 2: Data Structures

- Write a program to process a range or all elements in one- and two-dimensional arrays

Topic 3: Discrete Mathematics—Logic, Functions, and Sets

- Write programs that use simple and complex logic statements
- Write an algorithm that uses mathematical functions

Level II Objectives

Topic 2: Problem Solving

- Name and explain the steps in the problem-solving process
- Solve a problem by applying the problem-solving process

Topic 9: Ethical Issues

- Distinguish between ethical and legal issues
- Define intellectual property and state the impact of provisions to protect it
- Demonstrate behavior in the use of technology that conforms to school and local code

National Educational Technology Standards (NETS, 2007) Reflected in the Course Competencies

Creativity and Innovation—Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology

- Apply existing knowledge to generate new ideas, products, or processes
- Create original works as a means of personal or group expression
- Use models and simulations to explore complex systems and issues

Research and Information Fluency—Students apply digital tools to gather, evaluate, and use information

- Plan strategies to guide inquiry
- Process data and report results

Critical Thinking, Problem Solving, and Decision Making—Students use critical thinking skills to plan and conduct research, manage projects, solve problems, and make informed decisions using appropriate digital tools and resources

- Identify and define authentic problems and significant questions for investigation
- Plan and manage activities to develop a solution or complete a project
- Collect and analyze data to identify solutions and/or make informed decisions
- Use multiple processes and diverse perspective to explore alternative solutions

Digital Citizenship—Students understand human, cultural, and societal issues related to technology and practice legal and ethical behavior

- Advocate and practice safe, legal, and responsible use of information and technology
- Exhibit a positive attitude toward using technology that supports collaboration, learning, and productivity
- Exhibit leadership for digital citizenship
- Demonstrate a personal responsibility for lifelong learning

Technology Operations and Concepts—Students demonstrate a sound understanding of technology concepts, systems, and operations

- Understand and use technology systems
- Select and use applications effectively and productively

- Troubleshoot systems and applications
- Transfer current knowledge to learning of new technologies

Massachusetts Career/Vocational Technical Education Curriculum Frameworks Reflected in the Course Competencies

Arts and Communication Services Cluster—Design and Visual Communications Technical Knowledge and Skills

- 6.A Demonstrate proficiency in the use of computers and applications as well as an understanding of concepts underlying hardware, software, and connectivity
- 6.B Demonstrate responsible use of technology and an understanding of ethics and safety issues in using electronic media
- 6.C Demonstrate ability to use technology for research, problem solving, and communication

Information and Technology Services Cluster Technical Knowledge and Skills

2.A Apply Problem Solving and Troubleshooting Basics.

- Define and document a problem.
- Define possible causes of a problem.
- Determine and discuss possible solutions to a problem.
- Explain and perform basic troubleshooting and maintenance tasks.

Programming

2.D Explain programming concepts.

- Define what a computer program is.
- Define how a computer program runs.
- Identify the applications appropriate for each programming language.
- Define functions/methods/procedures.
- Define programming structures.
- Differentiate between procedural and object-oriented programming.
- Define purpose and use of flowcharting.

2.N Explain fundamental programming theory.

- Describe the relationship between hardware and software.
- Analyze programming languages for uses, structure, and environment.
- Classify the various programming languages by communication level.
- Summarize the function and operation of compilers and interpreters.
- List the stages of program development.
- Analyze a problem identifying desired outputs for given inputs.
- Describe the fundamental data types and their operations (including arrays).
- Design program logic using graphical techniques (flow charts).
- Design program logic using pseudocode techniques.
- Identify the use of program design tools.
- Explain structured/modular programming.

- Describe the information system (IS) life cycle.
- List the characteristics and uses of batch processing.
- List the characteristics and uses of interactive processing.
- List the characteristics and uses of event-driven, object-oriented processing.
- Illustrate characteristics of technical documentation associated with software development.

2.O Plan programs.

- Develop a problem statement.
- Define the assumptions that define the scope of the problem.
- List strategies used to gather known information.
- Apply known information to the problem statement.
- Hypothesize expected output.

2.P Develop programs.

- Develop programs using desired language.
- Develop programs that use arithmetic operations.
- Develop programs that use relational operators.
- Explain and apply the use of logical operators.
- Explain and apply compound conditions.
- Explain and apply control breaks.
- Explain and apply methods of calculating subtotals and final totals.
- Explain and apply iterative and conditional loops.
- Describe common development environments.
- Explain and apply the use of sort routines.
- Explain and apply the use of files in programming.
- Create sequential files.
- Create random files.
- Create, update, and delete records.
- Explain and apply the use of an array.
- Design and develop structures.
- Design and develop classes, subclasses.
- Apply language specific programming techniques.
- Test and run a program for desired output.
- Explain and apply methods used to debug a program.
- Utilize reference materials for problem solving.
- Generate executable code.

National Standards for Business Education Performance Standards Reflected in the Course Competencies (2007)

Information Technology

- Describe the impact of technology on the knowledge and skills needed for success in the workplace
- Explain how information technology has contributed to work productivity and team work

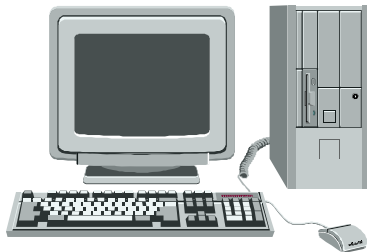
- Apply information technology skills to lifelong learning
- Describe interrelationships between hardware components and supportive software
- Access, navigate, and use online resources
- Interpret information for use in decision making
- Demonstrate legal and ethical behaviors when using information technologies
- Troubleshoot advanced hardware and supportive software problems

Technological Communication

- Demonstrate basic keyboarding skills and computer functions

Programming and Application Development

- Identify and define object-oriented programming
- Demonstrate the ability to code using object-oriented programming
- Identify and explain programming structures
- Differentiate between source and object code
- Identify and define the coding task
- Apply design principles to programming tasks
- Maintain and reengineer existing code
- Code common tasks (e.g., creating, adding, deleting, sorting, and updating records)
- Test, debug, and document code



Visual Basic Skills