Computer Programming Training from the World's Best-Selling Programming-Languages Textbook Authors

PRICE LIST COVER PAGE

AUTHORIZED
INFORMATION TECHNOLOGY SCHEDULE PRICELIST
GENERAL PURPOSE COMMERCIAL INFORMATION TECHNOLOGY
EQUIPMENT, SOFTWARE AND SERVICES

SIN 132-50 - TRAINING COURSES FOR INFORMATION TECHNOLOGY EQUIPMENT AND SOFTWARE (FPDS Code U012)

Deitel & Associates, Inc.
5 Clock Tower Place, Suite 450
Maynard, MA 01754
Phone: (978) 823-0130
http://www.deitel.com

Contract Number: _____
****will be furnished at time of award****

Period Covered by Contract: _____
****will be furnished at time of award****

General Services Administration
Federal Acquisition Service

Pricelist current through Modification #_____, dated _____.

Products and ordering information in this Authorized FSS Information Technology Schedule Pricelist are also available on the GSA Advantage! System. Agencies can browse GSA Advantage! by accessing the Federal Acquisition Service's Home Page via the Internet at http://www.fss.gsa.gov/

Table of Contents

Information for Ordering Activities

INFORMATION FOR ORDERING ACTIVITIES
APPLICABLE TO ALL SPECIAL ITEM NUMBERS

SPECIAL NOTICE TO AGENCIES: Small Business Participation

SBA strongly supports the participation of small business concerns in the Federal Acquisition Service. To enhance Small Business Participation SBA policy allows agencies to include in their procurement base and goals, the dollar value of orders expected to be placed against the Federal Supply Schedules, and to report accomplishments against these goals.

For orders exceeding the micropurchase threshold, FAR 8.404 requires agencies to consider the catalogs/pricelists of at least three schedule contractors or consider reasonably available information by using the GSA Advantage!™ on-line shopping service (www.fss.gsa.gov). The catalogs/pricelists, GSA Advantage!™ and the Federal Acquisition Service Home Page (www.fss.gsa.gov) contain information on a broad array of products and services offered by small business concerns.

This information should be used as a tool to assist ordering activities in meeting or exceeding established small business goals. It should also be used as a tool to assist in including small, small disadvantaged, and women-owned small businesses among those considered when selecting pricelists for a best value determination.

For orders exceeding the micropurchase threshold, customers are to give preference to small business concerns when two or more items at the same delivered price will satisfy their requirement.

1. GEOGRAPHIC SCOPE OF CONTRACT:

Domestic delivery is delivery within the 48 contiguous states, Alaska, Hawaii, Puerto Rico, Washington, DC, and U.S. Territories. Domestic delivery also includes a port or consolidation point, within the aforementioned areas, for orders received from overseas activities.

Overseas delivery is delivery to points outside of the 48 contiguous states, Washington, DC, Alaska, Hawaii, Puerto Rico, and U.S. Territories.

Offerors are requested to check one of the following boxes:

[ ] The Geographic Scope of Contract will be domestic and overseas delivery.
[ ] The Geographic Scope of Contract will be overseas delivery only.
[X] The Geographic Scope of Contract will be domestic delivery only.

For Special Item Number 132-53 Wireless Services ONLY, if awarded, list the limited geographic coverage area:

_____

2. CONTRACTOR'S ORDERING ADDRESS AND PAYMENT INFORMATION:

Deitel & Associates, Inc.
5 Clock Tower Place, Suite 450
Maynard, MA 01754
Phone: (978) 823-0130
http://www.deitel.com

Contractors are required to accept credit cards for payments equal to or less than the micro-purchase threshold for oral or written delivery orders. Credit cards will not be acceptable for payment above the micro-purchase threshold. In addition, bank account information for wire transfer payments will be shown on the invoice.

The following telephone number(s) can be used by ordering activities to obtain technical and/or ordering assistance:

(978) 823-0130

3. LIABILITY FOR INJURY OR DAMAGE

The Contractor shall not be liable for any injury to ordering activity personnel or damage to ordering activity property arising from the use of equipment maintained by the Contractor, unless such injury or damage is due to the fault or negligence of the Contractor.

4. STATISTICAL DATA FOR GOVERNMENT ORDERING OFFICE COMPLETION OF STANDARD FORM 279:

Block 9: G. Order/Modification Under Federal Schedule
Block 16: Data Universal Numbering System (DUNS) Number: 627688435
Block 30: Type of Contractor - B. Other Small Business

Block 31: Woman-Owned Small Business - No
Block 36: Contractor's Taxpayer Identification Number (TIN): 04-3255930

4a. CAGE Code: 1GWD0
4b. Contractor has registered with the Central Contractor Registration Database.

5. FOB DESTINATION

6. DELIVERY SCHEDULE

a. TIME OF DELIVERY: The Contractor shall deliver to destination within the number of calendar days after receipt of order (ARO), as set forth below:

SPECIAL ITEM NUMBER	DELIVERY TIME (Days ARO)
132-50	As required (and dependent on instructor availability)

b. URGENT REQUIREMENTS: When the Federal Supply Schedule contract delivery period does not meet the bona fide urgent delivery requirements of an ordering activity, ordering activities are encouraged, if time permits, to contact the Contractor for the purpose of obtaining accelerated delivery. The Contractor shall reply to the inquiry within 3 workdays after receipt. (Telephonic replies shall be confirmed by the Contractor in writing.) If the Contractor offers an accelerated delivery time acceptable to the ordering activity, any order(s) placed pursuant to the agreed upon accelerated delivery time frame shall be delivered within this shorter delivery time and in accordance with all other terms and conditions of the contract.

7. DISCOUNTS: Prices shown are NET Prices; Basic Discounts have been deducted.

a. Prompt Payment: .5% 15 days Net 30 days.
b. Quantity: None
c. Dollar Volume: None
d. Government Educational Institutions: Same discounts as all other Government customers.
e. Other

8. TRADE AGREEMENTS ACT OF 1979, as amended:

All items are U.S. made end products, designated country end products, Caribbean Basin country end products, Canadian end products, or Mexican end products as defined in the Trade Agreements Act of 1979, as amended.

9. STATEMENT CONCERNING AVAILABILITY OF EXPORT PACKING:

10. Small Requirements: The minimum dollar value of orders to be issued is $100.

11. MAXIMUM ORDER (All dollar amounts are exclusive of any discount for prompt payment.)

b. The Maximum Order value for the following Special Item Numbers (SINs) is $25,000:

Special Item Number 132-50 - Training Courses

12. ORDERING PROCEDURES FOR FEDERAL SUPPLY SCHEDULE CONTRACTS

Ordering activities shall use the ordering procedures of Federal Acquisition Regulation (FAR) 8.405 when placing an order or establishing a BPA for supplies or services. These procedures apply to all schedules.

a. FAR 8.405-1 Ordering procedures for supplies, and services not requiring a statement of work.

b. FAR 8.405-2 Ordering procedures for services requiring a statement of work.

13. FEDERAL INFORMATION TECHNOLOGY/TELECOMMUNICATION STANDARDS REQUIREMENTS: ordering activities acquiring products from this Schedule must comply with the provisions of the Federal Standards Program, as appropriate (reference: NIST Federal Standards Index). Inquiries to determine whether or not specific products listed herein comply with Federal Information Processing Standards (FIPS) or Federal Telecommunication Standards (FED-STDS), which are cited by ordering activities, shall be responded to promptly by the Contractor.

13.1 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATIONS (FIPS PUBS): Information Technology products under this Schedule that do not conform to Federal Information Processing Standards (FIPS) should not be acquired unless a waiver has been granted in accordance with the applicable "FIPS Publication." Federal Information Processing Standards Publications (FIPS PUBS) are issued by the U.S. Department of Commerce, National Institute of Standards and Technology (NIST), pursuant to National Security Act. Information concerning their availability and applicability should be obtained from the National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, Virginia 22161. FIPS PUBS include voluntary standards when these are adopted for Federal use. Individual orders for FIPS PUBS should be referred to the NTIS Sales Office, and orders for subscription service should be referred to the NTIS Subscription Officer, both at the above address, or telephone number (703) 487-4650.

13.2 FEDERAL TELECOMMUNICATION STANDARDS (FED-STDS): Telecommunication products under this Schedule that do not conform to Federal Telecommunication Standards (FED-STDS) should not be acquired unless a waiver has been granted in accordance with the applicable "FED-STD." Federal Telecommunication Standards are issued by the U.S. Department of Commerce, National Institute of Standards and Technology (NIST), pursuant to National Security Act. Ordering information and information concerning the availability of FED-STDS should be obtained from the GSA, Federal Acquisition Service, Specification Section, 470 East L'Enfant Plaza, Suite 8100, SW, Washington, DC 20407, telephone number (202)619-8925. Please include a self-addressed mailing label when requesting information by mail. Information concerning their applicability can be obtained by writing or calling the U.S. Department of Commerce, National Institute of Standards and Technology, Gaithersburg, MD 20899, telephone number (301)975-2833.

14. CONTRACTOR TASKS / SPECIAL REQUIREMENTS (C-FSS-370) (NOV 2001)

(a) Security Clearances: The Contractor may be required to obtain/possess varying levels of security clearances in the performance of orders issued under this contract. All costs associated with obtaining/possessing such security clearances should be factored into the price offered under the Multiple Award Schedule.

(b) Travel: The Contractor may be required to travel in performance of orders issued under this contract. Allowable travel and per diem charges are governed by Pub .L. 99-234 and FAR Part 31, and are reimbursable by the ordering agency or can be priced as a fixed price item on orders placed under the Multiple Award Schedule. The Industrial Funding Fee does NOT apply to travel and per diem charges.

NOTE: Refer to FAR Part 31.205-46 Travel Costs, for allowable costs that pertain to official company business travel in regards to this contract.

(c) Certifications, Licenses and Accreditations: As a commercial practice, the Contractor may be required to obtain/possess any variety of certifications, licenses and accreditations for specific FSC/service code classifications offered. All costs associated with obtaining/ possessing such certifications, licenses and accreditations should be factored into the price offered under the Multiple Award Schedule program.

(d) Insurance: As a commercial practice, the Contractor may be required to obtain/possess insurance coverage for specific FSC/service code classifications offered. All costs associated with obtaining/possessing such insurance should be factored into the price offered under the Multiple Award Schedule program.

(e) Personnel: The Contractor may be required to provide key personnel, resumes or skill category descriptions in the performance of orders issued under this contract. Ordering activities may require agency approval of additions or replacements to key personnel.

(f) Organizational Conflicts of Interest: Where there may be an organizational conflict of interest as determined by the ordering agency, the Contractor's participation in such order may be restricted in accordance with FAR Part 9.5.

(g) Documentation/Standards: The Contractor may be requested to provide products or services in accordance with rules, regulations, OMB orders, standards and documentation as specified by the agency's order.

(h) Data/Deliverable Requirements: Any required data/deliverables at the ordering level will be as specified or negotiated in the agency's order.

(i) Government-Furnished Property: As specified by the agency's order, the Government may provide property, equipment, materials or resources as necessary.

(j) Availability of Funds: Many Government agencies' operating funds are appropriated for a specific fiscal year. Funds may not be presently available for any orders placed under the contract or any option year. The Government's obligation on orders placed under this contract is contingent upon the availability of appropriated funds from which payment for ordering purposes can be made. No legal liability on the part of the Government for any payment may arise until funds are available to the ordering Contracting Officer.

15. CONTRACT ADMINISTRATION FOR ORDERING ACTIVITIES: Any ordering activity, with respect to any one or more delivery orders placed by it under this contract, may exercise the same rights of termination as might the GSA Contracting Officer under provisions of FAR 52.212-4, paragraphs (l) Termination for the ordering activity's convenience, and (m) Termination for Cause (See C.1.)

16. GSA ADVANTAGE!

GSA Advantage! is an on-line, interactive electronic information and ordering system that provides on-line access to vendors' schedule prices with ordering information. GSA Advantage! will allow the user to perform various searches across all contracts including, but not limited to:

(1) Manufacturer;
(2) Manufacturer's Part Number; and
(3) Product categories.

Agencies can browse GSA Advantage! by accessing the Internet World Wide Web utilizing a browser (ex.: NetScape). The Internet address is http://www.fss.gsa.gov/.

17. PURCHASE OF OPEN MARKET ITEMS

NOTE: Open Market Items are also known as incidental items, noncontract items, non-Schedule items, and items not on a Federal Supply Schedule contract. ODCs (Other Direct Costs) are not part of this contract and should be treated as open market purchases. Ordering Activities procuring open market items must follow FAR 8.402(f).

For administrative convenience, an ordering activity contracting officer may add items not on the Federal Supply Multiple Award Schedule (MAS) -- referred to as open market items -- to a Federal Supply Schedule blanket purchase agreement (BPA) or an individual task or delivery order, **only if**-

(1) All applicable acquisition regulations pertaining to the purchase of the items not on the Federal Supply Schedule have been followed (e.g., publicizing (Part 5), competition requirements (Part 6), acquisition of commercial items (Part 12), contracting methods (Parts 13, 14, and 15), and small business programs (Part 19));

(2) The ordering activity contracting officer has determined the price for the items not on the Federal Supply Schedule is fair and reasonable;

(3) The items are clearly labeled on the order as items not on the Federal Supply Schedule; and

(4) All clauses applicable to items not on the Federal Supply Schedule are included in the order.

18. CONTRACTOR COMMITMENTS, WARRANTIES AND REPRESENTATIONS

a. For the purpose of this contract, commitments, warranties and representations include, in addition to those agreed to for the entire schedule contract:

(1) Time of delivery/installation quotations for individual orders;

(2) Technical representations and/or warranties of products concerning performance, total system performance and/or configuration, physical, design and/or functional characteristics and capabilities of a product/equipment/ service/software package submitted in response to requirements which result in orders under this schedule contract.

(3) Any representations and/or warranties concerning the products made in any literature, description, drawings and/or specifications furnished by the Contractor.

b. The above is not intended to encompass items not currently covered by the GSA Schedule contract.

19. OVERSEAS ACTIVITIES

The terms and conditions of this contract shall apply to all orders for installation, maintenance and repair of equipment in areas listed in the pricelist outside the 48 contiguous states and the District of Columbia, except as indicated below:

_____

_____

Upon request of the Contractor, the ordering activity may provide the Contractor with logistics support, as available, in accordance with all applicable ordering activity regulations. Such ordering activity support will be provided on a reimbursable basis, and will only be provided to the Contractor's technical personnel whose services are exclusively required for the fulfillment of the terms and conditions of this contract.

20. BLANKET PURCHASE AGREEMENTS (BPAs)

The use of BPAs under any schedule contract to fill repetitive needs for supplies or services is allowable. BPAs may be established with one or more schedule contractors. The number of BPAs to be established is within the discretion of the ordering activity establishing the BPA and should be based on a strategy that is expected to maximize the effectiveness of the BPA(s). Ordering activities shall follow FAR 8.405-3 when creating and implementing BPA(s).

21. CONTRACTOR TEAM ARRANGEMENTS

Contractors participating in contractor team arrangements must abide by all terms and conditions of their respective contracts. This includes compliance with Clauses 552.238-74, Industrial Funding Fee and Sales Reporting, i.e., each contractor (team member) must report sales and remit the IFF for all products and services provided under its individual contract.

22. INSTALLATION, DEINSTALLATION, REINSTALLATION

The Davis-Bacon Act (40 U.S.C. 276a-276a-7) provides that contracts in excess of $2,000 to which the United States or the District of Columbia is a party for construction, alteration, or repair (including painting and decorating) of public buildings or public works with the United States, shall contain a clause that no laborer or mechanic employed directly upon the site of the work shall receive less than the prevailing wage rates as determined by the Secretary of Labor. The requirements of the Davis-Bacon Act do not apply if the construction work is incidental to the furnishing of supplies, equipment, or services. For example, the requirements do not apply to simple installation or alteration of a public building or public work that is incidental to furnishing supplies or equipment under a supply contract. However, if the construction, alteration or repair is segregable and exceeds $2,000, then the requirements of the Davis-Bacon Act applies.

The ordering activity issuing the task order against this contract will be responsible for proper administration and enforcement of the Federal labor standards covered by the Davis-Bacon Act. The proper Davis-Bacon wage determination will be issued by the ordering activity at the time a request for quotations is made for applicable construction classified installation, deinstallation, and reinstallation services under SIN 132-8.

23. SECTION 508 COMPLIANCE.

If applicable, Section 508 compliance information on the supplies and services in this contract are available in Electronic and Information Technology (EIT) at the following:

_____

The EIT standard can be found at: www.Section508.gov/.

24. PRIME CONTRACTOR ORDERING FROM FEDERAL SUPPLY SCHEDULES.

Prime Contractors (on cost reimbursement contracts) placing orders under Federal Supply Schedules, on behalf of an ordering activity, shall follow the terms of the applicable schedule and authorization and include with each order –

(a) A copy of the authorization from the ordering activity with whom the contractor has the prime contract (unless a copy was previously furnished to the Federal Supply Schedule contractor); and

(b) The following statement:

This order is placed under written authorization from _____ dated _____. In the event of any inconsistency between the terms and conditions of this order and those of your Federal Supply Schedule contract, the latter will govern.

25. INSURANCE—WORK ON A GOVERNMENT INSTALLATION (JAN 1997)(FAR 52.228-5)

(a) The Contractor shall, at its own expense, provide and maintain during the entire performance of this contract, at least the kinds and minimum amounts of insurance required in the Schedule or elsewhere in the contract.

(b) Before commencing work under this contract, the Contractor shall notify the Contracting Officer in writing that the required insurance has been obtained. The policies evidencing required insurance shall contain an endorsement to the effect that any cancellation or any material change adversely affecting the Government's interest shall not be effective—

(1) For such period as the laws of the State in which this contract is to be performed prescribe; or

(2) Until 30 days after the insurer or the Contractor gives written notice to the Contracting Officer, whichever period is longer.

(c) The Contractor shall insert the substance of this clause, including this paragraph (c), in subcontracts under this contract that require work on a Government installation and shall require subcontractors to provide and maintain the insurance required in the Schedule or elsewhere in the contract. The Contractor

shall maintain a copy of all subcontractors' proofs of required insurance, and shall make copies available to the Contracting Officer upon request.

26. SOFTWARE INTEROPERABILITY.

Offerors are encouraged to identify within their software items any component interfaces that support open standard interoperability. An item's interface may be identified as interoperable on the basis of participation in a Government agency-sponsored program or in an independent organization program. Interfaces may be identified by reference to an interface registered in the component registry located at http://www.core.gov.

27. ADVANCE PAYMENTS

A payment under this contract to provide a service or deliver an article for the United States Government may not be more than the value of the service already provided or the article already delivered. Advance or pre-payment is not authorized or allowed under this contract. (31 U.S.C. 3324)

TERMS AND CONDITIONS APPLICABLE TO PURCHASE OF TRAINING COURSES FOR GENERAL PURPOSE COMMERCIAL INFORMATION TECHNOLOGY EQUIPMENT AND SOFTWARE (SPECIAL ITEM NUMBER 132-50)

1. SCOPE

a. The Contractor shall provide training courses normally available to commercial customers, which will permit ordering activity users to make full, efficient use of general purpose commercial IT products. Training is restricted to training courses for those products within the scope of this solicitation.

b. The Contractor shall provide training at the ordering activity's location, as agreed to by the Contractor and the ordering activity.

2. ORDER

Written orders, EDI orders (GSA Advantage! and FACNET), credit card orders, and orders placed under blanket purchase agreements (BPAs) shall be the basis for the purchase of training courses in accordance with the terms of this contract. Orders shall include the student's name, course title, course date and time, and contracted dollar amount of the course.

3. TIME OF DELIVERY

The Contractor shall conduct training on the date (time, day, month, and year) agreed to by the Contractor and the ordering activity.

4. CANCELLATION AND RESCHEDULING

a. The ordering activity will notify the Contractor at least seventy-two (72) hours before the scheduled training date, if a student will be unable to attend. The Contractor will then permit the ordering activity to either cancel the order or reschedule the training at no additional charge. In the event the training class is rescheduled, the ordering activity will modify its original training order to specify the time and date of the rescheduled training class.

b. In the event the ordering activity fails to cancel or reschedule a training course within the time frame specified in paragraph a, above, the ordering activity will be liable for the contracted dollar amount of the training course.

c. The ordering activity reserves the right to substitute one student for another up to the first day of class.

d. In the event the Contractor is unable to conduct training on the date agreed to by the Contractor and the ordering activity, the Contractor must notify the ordering activity at least seventy-two (72) hours before the scheduled training date.

5. FOLLOW-UP SUPPORT

The Contractor agrees to provide each student with unlimited telephone support for a period of one (1) year from the completion of the training course. During this period, the student may contact the Contractor's instructors for refresher assistance and answers to related course curriculum questions.

6. PRICE FOR TRAINING

The price that the ordering activity will be charged will be the ordering activity training price in effect at the time of order placement, or the ordering activity price in effect at the time the training course is conducted, whichever is less.

7. INVOICES AND PAYMENT

Invoices for training shall be submitted by the Contractor after ordering activity completion of the training course. Charges for training must be paid in arrears (31 U.S.C. 3324). PROMPT PAYMENT DISCOUNT, IF APPLICABLE, SHALL BE SHOWN ON THE INVOICE.

8. FORMAT AND CONTENT OF TRAINING

a. The Ordering Activity is responsible for purchasing the books at its own expense directly from the publisher, Pearson. Pearson's GSA pricing and ordering information is available at http://www.pearsongovernmentsales.com. Such documentation will become the property of the student upon completion of the training class.

b. For hands-on training courses, there must be a one-to-one assignment of IT equipment to students; all equipment to be provided by the Ordering Activity and the Ordering Activity's expense.

c. The Contractor shall provide each student with a Certificate of Training at the completion of each training course.

d. The Contractor shall provide the following information for each training course offered:

(1) The course title and a brief description of the course content, to include the course format (e.g., lecture, discussion, hands-on training);

(2) The length of the course;

(3) Mandatory and desirable prerequisites for student enrollment;

(4) The minimum and maximum number of students per class;

(5) The locations where the course is offered;

(6) Class schedules; and

(7) Price (per student, per class (if applicable)).

e. For those courses conducted at the ordering activity's location, instructor travel charges (if applicable), including mileage and daily living expenses (e.g., per diem charges) are governed by Pub. L. 99-234 and FAR Part 31.205-46, and are reimbursable by the ordering activity on orders placed under the Multiple Award Schedule, as applicable, in effect on the date(s) the travel is performed. Contractors cannot use GSA city pair contracts. The Industrial Funding Fee does NOT apply to travel and per diem charges.

9. "NO CHARGE" TRAINING

The Contractor shall describe any training provided with equipment and/or software provided under this contract, free of charge, in the space provided below.

Deitel & Associates, Inc. General Terms and Conditions

1. Courses are presented on-site at the Ordering Activity's location.
2. The Ordering Activity must provide:
    a. A lecture room with whiteboards and an overhead projector that can be connected to the instructor's notebook computer.
    b. Computers with appropriate software that students will use to complete the lab exercises. Deitel will provide instructions and web links for any software installations required for the course. It is the Ordering Activity's responsibility to purchase (if necessary) at its own expense and install the software prior to the start of the course.
    c. Appropriate Internet access (if required for the course).
3. The maximum number of students per class is 20. There is no minimum.
4. The Ordering Activity is responsible for purchasing the books at its own expense directly from the publisher, Pearson. Please contact:

    Kathryn Bass, Government Sales Executive
    Pearson Education
    Phone: 703-404-9194
    Fax 703-404-9195
    Kathryn.Bass@Pearson.com
    GSA Purchasing and Ordering Information: www.pearsongovernmentsales.com

5. Order Activity must provide a purchase order stating:
    a. The class titles.
    b. The length of the course in days and the specific dates the class will be offered, as agreed upon between the Ordering Activity and Deitel.
    c. The lecture fee per course (for up to 20 students maximum per course).
    d. That the Ordering Activity will order the books at its own expense directly from Pearson (the publisher).
    e. That instructor travel will be reimbursed, as per government guidelines.
    f. The payment terms for each individual course: 0.5% discount 15 days Net 30 days from the completion of each course.

SIN 132-50 Training Courses

Java™ Programming

- Java101—Introduction to Java for Non-Programmers: Part 1
- Java102—Introduction to Java for Non-Programmers: Part 2
- Java112—Java for Visual Basic, C or COBOL Programmers
- Java200—Java for C++ or C# Programmers
- Java300—Advanced Java

C++ Programming

- C++101—Introduction to C++ for Non-Programmers: Part 1
- C++102—Introduction to C++ for Non Programmers: Part 2
- C++200—C++ and Object Oriented Programming

C Programming

- C101—Introduction to C for Non-Programmers: Part 1
- C102—Introduction to C for Non-Programmers: Part 2
- C200—C for Programmers

Visual C#® 2008 Programming

- VC#101—Introduction to Visual C# 2008 for Non-Programmers: Part 1
- VC#102—Introduction to Visual C# 2008 for Non-Programmers: Part 2
- VC#112—Visual C# 2008 for Visual Basic, C or COBOL Programmers
- VC#200—Visual C# 2008 for Java or C++ Programmers
- VC#300—Advanced Visual C# 2008

Visual Basic® 2008 Programming

- VB101—Introduction to Visual Basic 2008 for Non-Programmers: Part 1
- VB102—Introduction to Visual Basic 2008 for Non-Programmers: Part 2
- VB112—Visual Basic 2008 for VB6, C or COBOL Programmers
- VB200—Visual Basic 2008 for Java, C# or C++ Programmers
- VB300—Advanced Visual Basic 2008

Visual C++ 2008 Programming

- VC++101—Introduction to Visual C++ 2008 for Non-Programmers: Part 1
- VC++102—Introduction to Visual C++ 2008 for Non Programmers: Part 2
- VC++200—Visual C++ 2008 and Object Oriented Programming

Internet and Web Programming

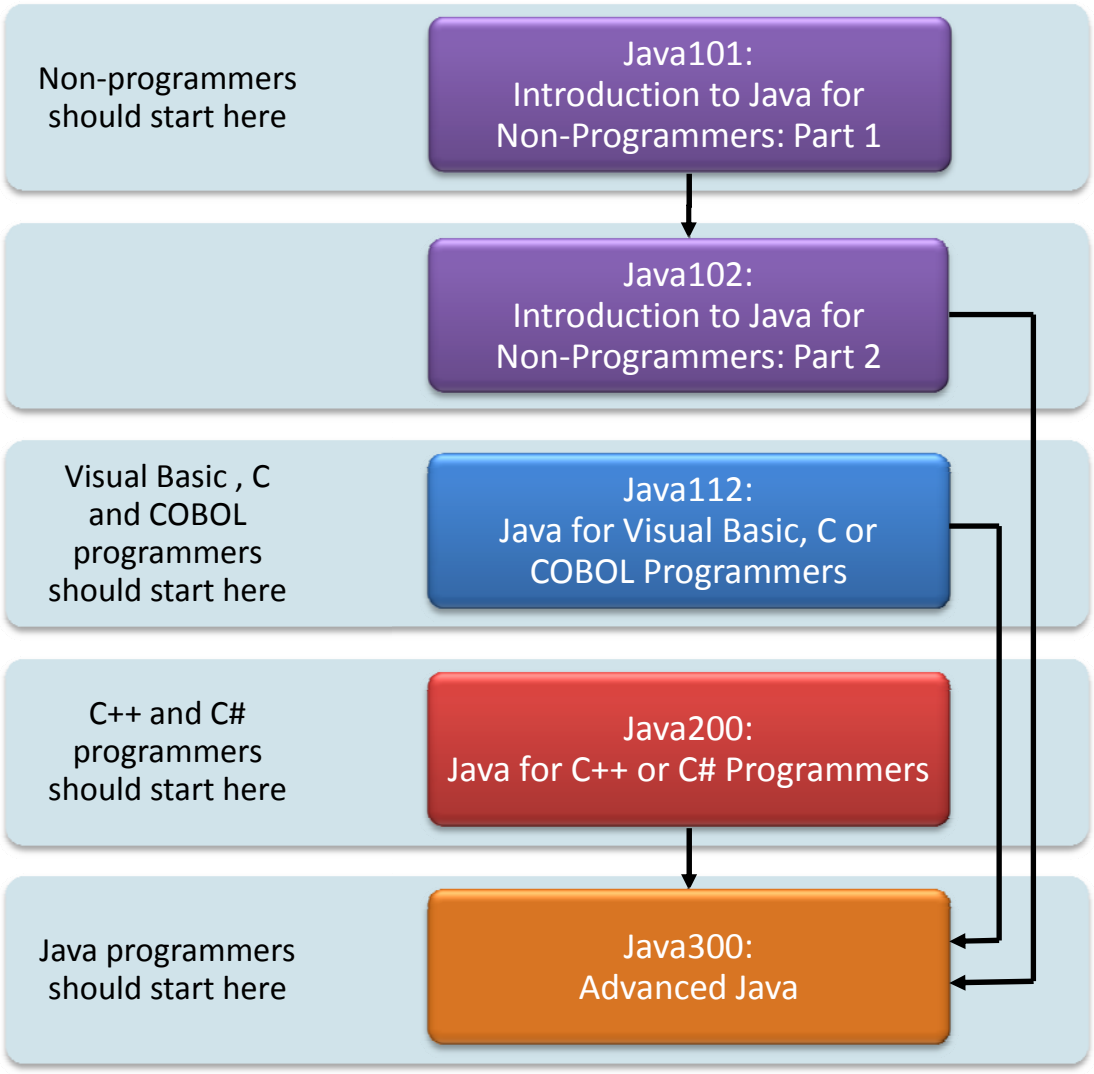
- Web201—Client-side Internet and Web Programming
- Web202—Rich Internet Application (RIA) Development
- Web203—Server-side Internet and Web Programming

Java™ Programming Curriculum

Overview

Java has become the language of choice for implementing Internet-based applications and software for devices that communicate over a network. There are now billions of Java-enabled computers and devices worldwide. Java is popular in the large-scale applications arena and is the preferred language for meeting many organizations' enterprise-wide programming needs. Java is now the most widely used programming language. These courses help novices become literate in Java programming and help experienced programmers become skilled Java developers.

Deitel & Associates, Inc. is the world's leading programming language textbook/professional book authors and our courses are based on our best-selling books.



Java101—Introduction to Java for Non-Programmers: Part 1

Overview
This introductory 5-day, lecture-and-lab course teaches people with little or no programming experience how to program with the Java programming language. Topics include introductory object-oriented programming, algorithmic thinking, problem solving, control statements, primitive types, operators, keyboard input, screen output, methods (user-defined and API), arrays and strings. The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. After taking this course, students will be prepared to take Java201—*Introduction to Java for Non-Programmers: Part 2*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Our introductory classes emphasize achieving program clarity through the proven software-development techniques. Nonprogrammers learn skills that underlie good programming through dozens of complete, working programs—we call this the live-code approach. This hands-on approach rapidly builds the confidence of new programmers, because all programming concepts are presented in the context in which they will be used.

Prerequisites
- Little or no previous programming experience
- General familiarity with your operating system environment
- Ability to create and edit text files

Introduction
- Overview of *Java How to Program, 7/e*
- Java Standard Edition (Java SE) 6
- Tools for programming in Java

Introduction to Java Applications
- A First Program in Java: Printing a Line of Text
- Modifying Our First Java Program
- Displaying Text with `printf`
- Another Java Application: Adding Integers
- Decision Making: Equality and Relational Operators

Introduction to Classes and Objects
- Classes, Objects, Methods and Instance Variables
- Declaring a Class with a Method and Instantiating an Object of a Class
- Declaring a Method with a Parameter
- Instance Variables, *set* Methods and *get* Methods
- Primitive Types vs. Reference Types
- Initializing Objects with Constructors
- Floating-Point Numbers and Type double

Control Statements: Part 1
- `if` Single-Selection Statement
- `if` … `else` Double-Selection Statement
- `while` Repetition Statement
- Formulating Algorithms: Counter-Controlled Repetition
- Formulating Algorithms: Sentinel-Controlled Repetition
- Formulating Algorithms: Nested Control Statements
- Compound Assignment Operators

- Increment and Decrement Operators
- Primitive Types

Control Statements: Part 2
- Essentials of Counter-Controlled Repetition
- `for` Repetition Statement
- Examples Using the `for` Statement
- `do` … `while` Repetition Statement
- `switch` Multiple-Selection Statement
- `break` and `continue` Statements
- Logical Operators

Methods: A Deeper Look
- `static` Methods, `static` Fields and Class `Math`
- Declaring Methods with Multiple Parameters
- Notes on Declaring and Using Methods
- Method-Call Stack and Activation Records
- Argument Promotion and Casting
- Java API Packages
- Case Study: Random-Number Generation
- Generalized Scaling and Shifting of Random Numbers
- Case Study: A Game of Chance (Introducing Enumerations)
- Scope of Declarations
- Method Overloading

Arrays
- Declaring and Creating Arrays
- Examples Using Arrays
- Case Study: Card Shuffling and Dealing Simulation
- Enhanced `for` Statement
- Passing Arrays to Methods
- Case Study: Class `GradeBook` Using an Array to Store Grades
- Multidimensional Arrays
- Case Study: Class `GradeBook` Using a Two-Dimensional Array

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
  - Materials: *Java How to Program, 7/e* (ISBN: 0132222205)
  - To order the books at the GSA rates, please contact:
    Kathryn Bass, Government Sales Executive
    Pearson Education
    Phone: 703-404-9194
    Fax 703-404-9195
    Kathryn.Bass@Pearson.com
    GSA Purchases: www.pearsongovernmentsales.com
    Pearson GSA Contract # GS14F8023A

Java102—Introduction to Java for Non-Programmers: Part 2

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with the Java programming language. Key topics include the Java Development Kit (JDK), Java SE 6, classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, packages, event-driven programming, graphical user interfaces (GUIs), exception handling, file processing and generic collections. Students create Java applications, understand Java object-oriented programming, learn to use various Java Application Programming Interfaces (APIs) and participate in extensive hands-on laboratory assignments. After taking this course and gaining some practical Java programming experience, students will be prepared to take Java300—*Advanced Java*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- Java101—*Java for Non-Programmers*, or
- Ability to program in a high-level language other than C/C++

Introduction

Review of Java syntax and concepts presented in the following chapters of *Java How to Program, 7/e*:

- Chapter 2—Introduction to Java Applications
- Chapter 3—Introduction to Classes and Objects
- Chapter 4—Control Statements: Part 1
- Chapter 5—Control Statements: Part 2
- Chapter 6—Methods: A Deeper Look
- Chapter 7—Arrays

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Controlling Access to Members
- Referring to the Current Object's Members with the `this` Reference
- `Time` Class Case Study: Overloaded Constructors
- Default and No-Argument Constructors
- Composition
- Enumerations
- Garbage Collection and Method `finalize`
- `static` Class Members
- `static` Import
- `final` Instance Variables
- `Time` Class Case Study: Creating Packages

Object-Oriented Programming: Inheritance

- Superclasses and Subclasses
- `protected` Members
- Relationship between Superclasses and Subclasses
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy Using `protected` Instance Variables
- `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy Using `private` Instance Variables

- Constructors in Subclasses
- Software Engineering with Inheritance

Object-Oriented Programming: Polymorphism

- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Using Polymorphism
- Demonstrating Polymorphic Processing, Operator `instanceof` and Downcasting
- Summary of the Allowed Assignments Between Superclass and Subclass Variables
- `final` Methods and Classes
- Case Study: Creating and Using Interfaces
- Developing a `Payable` Hierarchy
- Declaring Constants with Interfaces
- Common Interfaces of the Java API

GUI Components: Part 1

- Simple GUI-Based Input/Output with `JOptionPane`
- Overview of Swing Components
- Displaying Text and Images in a Window
- Text Fields and an Introduction to Event Handling with Nested Classes
- Common GUI Event Types and Listener Interfaces
- How Event Handling Works
- `JButton` component
- Buttons That Maintain State
- `JCheckBox` component
- `JRadioButton` component
- `JComboBox` and Using an Anonymous Inner Class for Event Handling
- `JList` component
- Mouse Event Handling and Adapter Classes
- `JPanel` Subclass for Drawing with the Mouse
- Layout Managers
- Using Panels to Manage More Complex Layouts

Exception Handling

- Exception-Handling Overview
- Example: Divide by Zero without Exception Handling
- Example: Handling `ArithmeticException`s and `InputMismatchException`s
- When to Use Exception Handling
- Java Exception Hierarchy
- `finally` Block
- Stack Unwinding
- `printStackTrace` , `getStackTrace` and `getMessage`
- Chained Exceptions
- Declaring New Exception Types
- Assertions

Files and Streams

- Data Hierarchy
- Files and Streams
- Class `File`

- Sequential-Access Text Files
- Creating a Sequential-Access Text File
- Reading Data from a Sequential-Access Text File
- Updating Sequential-Access Files
- Object Serialization
- Creating a Sequential-Access File Using Object Serialization
- Reading and Deserializing Data from a Sequential-Access File
- Additional `java.io` Classes
- Opening Files with `JFileChooser`

Collections

- Collections Overview
- Class `Arrays`
- Interface `Collection` and Class `Collections`
- `ArrayList` class and `Iterators`
- `LinkedList` class
- Collections Algorithms
- Algorithm `sort`
- Algorithm `shuffle`
- Algorithms `reverse`, `fill`, `copy`, `max` and `min`
- Algorithm `binarySearch`
- Stack Class of Package `java.util`
- Class `PriorityQueue` and Interface `Queue`
- Sets
- Maps

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Java How to Program, 7/e* (ISBN: 0132222205)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Java112—Java for Visual Basic, C or COBOL Programmers

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with the Java programming language. Key topics include the Java Development Kit (JDK), Java SE 6, classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, packages, event-driven programming, graphical user interfaces (GUIs), exception handling, file processing and generic collections. Students create Java applications, understand Java object-oriented programming, learn to use various Java Application Programming Interfaces (APIs) and participate in extensive hands-on laboratory assignments. After taking this course, students will be prepared to take Java300—*Advanced Java*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- Programming experience in Visual Basic, C or COBOL

Introduction

Review of Java syntax and concepts presented in the following chapters of *Java How to Program, 7/e*:

- Chapter 2—Introduction to Java Applications
- Chapter 3—Introduction to Classes and Objects
- Chapter 4—Control Statements: Part 1
- Chapter 5—Control Statements: Part 2
- Chapter 6—Methods: A Deeper Look
- Chapter 7—Arrays

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Controlling Access to Members
- Referring to the Current Object's Members with the `this` Reference
- `Time` Class Case Study: Overloaded Constructors
- Default and No-Argument Constructors
- Composition
- Enumerations
- Garbage Collection and Method `finalize`
- `static` Class Members
- `static` Import
- `final` Instance Variables
- `Time` Class Case Study: Creating Packages

Object-Oriented Programming: Inheritance

- Superclasses and Subclasses
- `protected` Members
- Relationship between Superclasses and Subclasses
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy Using `protected` Instance Variables
- `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy Using `private` Instance Variables
- Constructors in Subclasses
- Software Engineering with Inheritance

Object-Oriented Programming: Polymorphism

- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Using Polymorphism
- Demonstrating Polymorphic Processing, Operator `instanceof` and Downcasting
- Summary of the Allowed Assignments Between Superclass and Subclass Variables
- `final` Methods and Classes
- Case Study: Creating and Using Interfaces
- Developing a `Payable` Hierarchy
- Declaring Constants with Interfaces
- Common Interfaces of the Java API

GUI Components: Part 1

- Simple GUI-Based Input/Output with `JOptionPane`
- Overview of Swing Components
- Displaying Text and Images in a Window
- Text Fields and an Introduction to Event Handling with Nested Classes
- Common GUI Event Types and Listener Interfaces
- How Event Handling Works
- `JButton` component
- Buttons That Maintain State
- `JCheckBox` component
- `JRadioButton` component
- `JComboBox` and Using an Anonymous Inner Class for Event Handling
- `JList` component
- Mouse Event Handling and Adapter Classes
- `JPanel` Subclass for Drawing with the Mouse
- Layout Managers
- Using Panels to Manage More Complex Layouts

Exception Handling

- Exception-Handling Overview
- Example: Divide by Zero without Exception Handling
- Example: Handling `ArithmeticExceptions` and `InputMismatchExceptions`
- When to Use Exception Handling
- Java Exception Hierarchy
- `finally` Block
- Stack Unwinding
- `printStackTrace` , `getStackTrace` and `getMessage`
- Chained Exceptions
- Declaring New Exception Types
- Assertions

Files and Streams

- Data Hierarchy
- Files and Streams
- Class `File`
- Sequential-Access Text Files
- Creating a Sequential-Access Text File
- Reading Data from a Sequential-Access Text File

- Updating Sequential-Access Files
- Object Serialization
- Creating a Sequential-Access File Using Object Serialization
- Reading and Deserializing Data from a Sequential-Access File
- Additional `java.io` Classes
- Opening Files with `JFileChooser`

Collections
- Collections Overview
- Class `Arrays`
- Interface `Collection` and Class `Collections`
- `ArrayList` class and `Iterators`
- `LinkedList` class
- Collections Algorithms
- Algorithm `sort`
- Algorithm `shuffle`
- Algorithms `reverse`, `fill`, `copy`, `max` and `min`
- Algorithm `binarySearch`
- Stack Class of Package `java.util`
- Class `PriorityQueue` and Interface `Queue`
- Sets
- Maps

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study
An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Java How to Program, 7/e* (ISBN: 0132222205)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Java200—Java for C++ or C# Programmers

Overview
This 5-day, lecture-and-lab course presents object-oriented programming with the Java programming language. Key topics include the Java Development Kit (JDK), Java SE 6, classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, packages, event-driven programming, graphical user interfaces (GUIs), exception handling, file processing, multithreading, generics and generic collections. Students create Java applications, understand Java object-oriented programming, learn to use various Java Application Programming Interfaces (APIs) and participate in extensive hands-on laboratory assignments. After taking this course, students will be prepared to take Java300—*Advanced Java*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- Ability to program in C++ or C#

Introduction
Review of Java syntax and concepts presented in the following chapters of *Java How to Program, 7/e*:
- Chapter 2—Introduction to Java Applications
- Chapter 3—Introduction to Classes and Objects
- Chapter 4—Control Statements: Part 1
- Chapter 5—Control Statements: Part 2
- Chapter 6—Methods: A Deeper Look
- Chapter 7—Arrays

Classes and Objects: A Deeper Look
- `Time` Class Case Study
- Controlling Access to Members
- Referring to the Current Object's Members with the `this` Reference
- `Time` Class Case Study: Overloaded Constructors
- Default and No-Argument Constructors
- Composition
- Enumerations
- Garbage Collection and Method `finalize`
- `static` Class Members
- `static` Import
- `final` Instance Variables
- `Time` Class Case Study: Creating Packages

Object-Oriented Programming: Inheritance
- Superclasses and Subclasses
- `protected` Members
- Relationship between Superclasses and Subclasses
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy Using `protected` Instance Variables
- `CommissionEmployee` – `BasePlusCommissionEmployee` Inheritance Hierarchy Using `private` Instance Variables
- Constructors in Subclasses
- Software Engineering with Inheritance

Object-Oriented Programming: Polymorphism

- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Using Polymorphism
- Demonstrating Polymorphic Processing, Operator `instanceof` and Downcasting
- Summary of the Allowed Assignments Between Superclass and Subclass Variables
- `final` Methods and Classes
- Case Study: Creating and Using Interfaces
- Developing a `Payable` Hierarchy
- Declaring Constants with Interfaces
- Common Interfaces of the Java API

GUI Components: Part 1

- Simple GUI-Based Input/Output with `JOptionPane`
- Overview of Swing Components
- Displaying Text and Images in a Window
- Text Fields and an Introduction to Event Handling with Nested Classes
- Common GUI Event Types and Listener Interfaces
- How Event Handling Works
- `JButton` component
- Buttons That Maintain State
- `JCheckBox` component
- `JRadioButton` component
- `JComboBox` and Using an Anonymous Inner Class for Event Handling
- `JList` component
- Mouse Event Handling and Adapter Classes
- `JPanel` Subclass for Drawing with the Mouse
- Layout Managers
- Using Panels to Manage More Complex Layouts

Exception Handling

- Exception-Handling Overview
- Example: Divide by Zero without Exception Handling
- Example: Handling `ArithmeticExceptions` and `InputMismatchExceptions`
- When to Use Exception Handling
- Java Exception Hierarchy
- `finally` Block
- Stack Unwinding
- `printStackTrace` , `getStackTrace` and `getMessage`
- Chained Exceptions
- Declaring New Exception Types
- Assertions

Files and Streams

- Data Hierarchy
- Files and Streams
- Class `File`
- Sequential-Access Text Files
- Creating a Sequential-Access Text File
- Reading Data from a Sequential-Access Text File

- Updating Sequential-Access Files
- Object Serialization
- Creating a Sequential-Access File Using Object Serialization
- Reading and Deserializing Data from a Sequential-Access File
- Additional `java.io` Classes
- Opening Files with `JFileChooser`

Generics

- Motivation for Generic Methods
- Generic Methods: Implementation and Compile-Time Translation
- Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type
- Overloading Generic Methods
- Generic Classes
- Raw Types
- Wildcards in Methods That Accept Type Parameters

Collections

- Collections Overview
- Class `Arrays`
- Interface `Collection` and Class `Collections`
- `ArrayList` class and `Iterator`s
- `LinkedList` class
- Collections Algorithms
- Algorithm `sort`
- Algorithm `shuffle`
- Algorithms `reverse`, `fill`, `copy`, `max` and `min`
- Algorithm `binarySearch`
- Stack Class of Package `java.util`
- Class `PriorityQueue` and Interface `Queue`
- Sets
- Maps

Multithreading

- Thread States: Life Cycle of a Thread
- Thread Priorities and Thread Scheduling
- Creating and Executing Threads
- Runnable s and the Thread Class
- Thread Management with the Executor Framework
- Thread Synchronization
- Unsynchronized Data Sharing
- Synchronized Data Sharing—Making Operations Atomic
- Producer/Consumer Relationship without Synchronization
- Producer/Consumer Relationship: `ArrayBlockingQueue`
- Producer/Consumer Relationship with Synchronization
- Producer/Consumer Relationship: Bounded Buffers
- Producer/Consumer Relationship: The `Lock` and `Condition` Interfaces
- Multithreading with GUI: Performing Computations in a Worker Thread
- Multithreading with GUI: Processing Intermediate Results with `SwingWorker`

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study
An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Java How to Program, 7/e* (ISBN: 0132222205)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Java300—Advanced Java

Overview

This 5-day, lecture-and-lab course presents database-intensive, desktop- and web-application development techniques in Java. Many of the topics are presented in the context of the Netbeans IDE (www.netbeans.com) and use MySQL/Java DB for the database and Sun's Java System Application Server to host the web applications. Many of the techniques presented in this course can be applied to other IDE, databases and application servers. Key topics include networking, JDBC database access, JavaServer Faces (JSF), Ajax-enabled JSF components, building web services and consuming web services. Students will participate in extensive hands-on laboratory assignments. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- Java200—*Java for C++ or C# Programmers*, or
- Java112—*Java for Visual Basic, C or COBOL Programmers*, or
- Java 102—*Introduction to Java for Non-Programmers: Part 2*, plus practical Java programming experience, or
- Equivalent Java programming experience

Networking

- Manipulating URLs
- Reading a File on a Web Server
- Establishing a Simple Server Using Stream Sockets
- Establishing a Simple Client Using Stream Sockets
- Client/Server Interaction with Stream Socket Connections
- Connectionless Client/Server Interaction with Datagrams
- Client/Server Tic-Tac-Toe Using a Multithreaded Server

Accessing Databases with JDBC

- Relational Databases
- Relational Database Overview: The `books` Database
- SQL
- Basic `SELECT` Query
- `WHERE` Clause
- `ORDER BY` Clause
- Merging Data from Multiple Tables: `INNER JOIN`
- `INSERT` Statement
- `UPDATE` Statement
- `DELETE` Statement
- Instructions for installing MySQL and MySQL Connector/J
- Instructions for Setting Up a MySQL User Account
- Creating Database books in MySQL
- Manipulating Databases with JDBC
- Connecting to and Querying a Database
- Querying the books Database
- `RowSet` Interface
- Java DB/Apache Derby
- `PreparedStatements`
- Stored Procedures
- Transaction Processing

Web Applications: Part 1

- Downloading, Installing and Configuring Netbeans and the Sun Java System Application Server
- Simple HTTP Transactions
- Multitier Application Architecture
- Java Web Technologies
- Servlets
- JavaServer Pages
- JavaServer Faces
- Web Technologies in Netbeans
- Creating and Running a Simple Application in Netbeans
- Examining a JSP File
- Examining a Page Bean File
- Event-Processing Life Cycle
- Relationship Between the JSP and Page Bean Files
- Examining the XHTML Generated by a Java Web Application
- Building a Web Application in Netbeans
- JSF Components
- Text and Graphics Components
- Validation Using Validator Components and Custom Validators
- Session Tracking
- Cookies
- Session Tracking with the `SessionBean` Object

Web Applications: Part 2

- Accessing Databases in Web Applications
- Building a Web Application that Displays Data from a Database
- Modifying the Page Bean File for the `AddressBook` Application
- Ajax-Enabled JSF Components
- Java BluePrints Component Library
- `AutoComplete` Text Field and Virtual Forms
- Configuring Virtual Forms
- JSP File with Virtual Forms and an `AutoComplete` Text Field
- Providing Suggestions for an `AutoComplete` Text Field
- Using the Google Maps Map Viewer Component in a Web Application

JAX-WS Web Services

- Java Web Services Basics
- Creating, Publishing, Testing and Describing a Web Service
- Creating a Web Application Project and Adding a Web Service Class in Netbeans
- Defining the `HugeInteger` Web Service in Netbeans
- Publishing the `HugeInteger` Web Service from Netbeans
- Testing the `HugeInteger` Web Service with Sun Java System Application Server's Tester Web page
- Describing a Web Service with the Web Service Description Language (WSDL)
- Consuming a Web Service
- Creating a Client in Netbeans to Consume the `HugeInteger` Web Service
- Consuming the `HugeInteger` Web Service
- SOAP
- Session Tracking in Web Services
- Creating a `Blackjack` Web Service
- Consuming the `Blackjack` Web Service

- Consuming a Database-Driven Web Service from a Web Application
- Configuring Java DB in Netbeans and Creating the Reservation Database
- Creating a Web Application to Interact with the Reservation Web Service
- Passing an Object of a User-Defined Type to a Web Service

Price

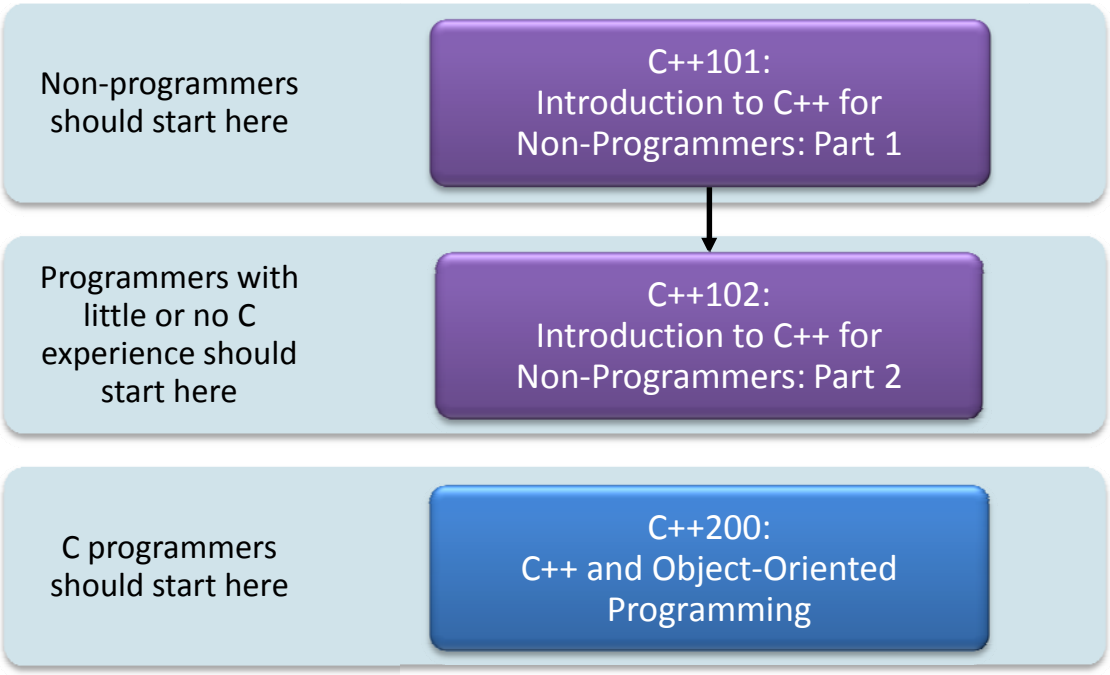
- $13,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Java How to Program, 7/e* (ISBN: 0132222205)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

C++ Programming Curriculum

Overview

C++ is a powerful computer programming language that is appropriate for technically oriented people with little or no programming experience and for experienced programmers to use in building substantial information systems. C++ is popular for developing high-performance software for operating systems, real-time systems, embedded systems and more.

Deitel & Associates, Inc. is the world's leading programming language textbook/professional book authors and our courses are based on our best-selling books.



C++101—Introduction to C++ for Non-Programmers: Part 1

Overview
This 5-day, lecture-and-lab course teaches fundamental C++ concepts such as algorithmic thinking, problem solving, introduction to classes and objects, control structures (`if`, `if…else`, `switch`, `while`, `do…while`, `for`), data types, operators, input/output, functions (user-defined and library), single-subscripted arrays and strings. After taking this course, students will be prepared to take C++102—*Introduction to C++ for Non-Programmers: Part 2*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Our introductory classes emphasize achieving program clarity through the proven software-development techniques. Nonprogrammers learn skills that underlie good programming through dozens of complete, working programs—we call this the live-code approach. This hands-on approach rapidly builds the confidence of new programmers, because all programming concepts are presented in the context in which they will be used.

Prerequisites
- General familiarity with your operating system environment
- Ability to create and edit text files

Introduction to C++ Programming
- First Program in C++: Printing a Line of Text
- Modifying Our First C++ Program
- Another C++ Program: Adding Integers
- Arithmetic
- Decision Making: Equality and Relational Operators

Introduction to Classes and Objects
- Classes, Objects, Member Functions and Data Members
- Overview of the Chapter Examples
- Defining a Class with a Member Function
- Defining a Member Function with a Parameter
- Data Members, *set* Functions and *get* Functions
- Initializing Objects with Constructors
- Placing a Class in a Separate File for Reusability
- Separating Interface from Implementation
- Validating Data with *set* Functions

Control Statements: Part 1
- Algorithms
- Pseudocode
- Control Structures
- `if` Selection Statement
- `if…else` Double-Selection Statement
- `while` Repetition Statement
- Formulating Algorithms: Counter-Controlled Repetition
- Formulating Algorithms: Sentinel-Controlled Repetition
- Formulating Algorithms: Nested Control Statements
- Assignment Operators
- Increment and Decrement Operators

Control Statements: Part 2

- Essentials of Counter-Controlled Repetition
- `for` Repetition Statement
- Examples Using the `for` Statement
- `do…while` Repetition Statement
- `switch` Multiple-Selection Statement
- `break` and `continue` Statements
- Logical Operators
- Confusing the Equality (==) and Assignment (=) Operators

Functions

- Math Library Functions
- Function Definitions with Multiple Parameters
- Function Prototypes and Argument Coercion
- C++ Standard Library Header Files
- Case Study: Random Number Generation
- Case Study: Game of Chance; Introducing `enum`
- Storage Classes
- Scope Rules
- Function Call Stack and Activation Records

Arrays

- Arrays
- Declaring Arrays
- Examples Using Arrays
- Declaring an Array and Using a Loop to Initialize the Array's Elements
- Initializing an Array in a Declaration with an Initializer List
- Specifying an Array's Size with a Constant Variable and Setting Array Elements with Calculations
- Summing the Elements of an Array
- Using Bar Charts to Display Array Data Graphically
- Using the Elements of an Array as Counters
- Using Arrays to Summarize Survey Results
- Using Character Arrays to Store and Manipulate Strings

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *C++ How to Program, 6/e* (ISBN: 0136152503)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

C++102—Introduction to C++ for Non Programmers: Part 2

Overview
This 5-day, lecture-and-lab course covers in-line functions, overloaded functions, function templates, default arguments, references, const qualifier, arrays, multiple-subscripted arrays, pointers, pointer arithmetic, function pointers, strings, optimizing memory use and improving execution performance. The course can be customized to the Ordering Activity's needs at no additional charge.

Prerequisites
- C++101—*Introduction to C++ for Non-Programmers: Part 1* or equivalent programming experience

Introduction
Review of C++ syntax and concepts presented in the following chapters of *C++ How to Program, 6/e* as part of the course *Introduction to C++ for Non-Programmers: Part 1*.
- Chapter 2—Introduction to C++ Programming
- Chapter 3—Introduction to Classes and Objects
- Chapter 4—Control Statements: Part 1
- Chapter 5—Control Statements: Part 2
- Chapter 6—Functions
- Chapter 7—Arrays

Functions
C++ as a better C
- Functions with Empty Parameter Lists
- Inline Functions
- References and Reference Parameters
- Default Arguments
- Unary Scope Resolution Operator
- Function Overloading
- Function Templates

Arrays and Vectors
- Static Local Arrays and Automatic Local Arrays
- Passing Arrays to Functions
- Case Study: Class GradeBook Using an Array to Store Grades
- Searching Arrays with Linear Search
- Sorting Arrays with Insertion Sort
- Multidimensional Arrays
- Case Study: Class GradeBook Using a Two-Dimensional Array
- Introduction to C++ Standard Library Class Template vector

Pointers and Pointer-Based Strings
- Pointer Variable Declarations and Initialization
- Pointer Operators
- Passing Arguments to Functions by Reference with Pointers
- Using const with Pointers
- Selection Sort Using Pass-by-Reference
- `sizeof` Operator
- Pointer Expressions and Pointer Arithmetic
- Relationship Between Pointers and Arrays
- Arrays of Pointers
- Case Study: Card Shuffling and Dealing Simulation

- Function Pointers
- Introduction to Pointer-Based String Processing
- Fundamentals of Characters and Pointer-Based Strings
- String-Manipulation Functions of the String-Handling Library

Classes: A Deeper Look, Part 1
- `Time` Class Case Study
- Class Scope and Accessing Class Members
- Separating Interface from Implementation
- Access Functions and Utility Functions
- `Time` Class Case Study: Constructors with Default Arguments
- Destructors
- When Constructors and Destructors Are Called
- `Time` Class Case Study: A Subtle Trap—Returning a Reference to a private Data Member
- Default Memberwise Assignment

Classes: A Deeper Look, Part 2
- `const` (Constant) Objects and const Member Functions
- Composition: Objects as Members of Classes
- `friend` Functions and friend Classes
- Using the `this` Pointer
- Dynamic Memory Management with Operators new and delete
- `static` Class Members
- Data Abstraction and Information Hiding
- Proxy Classes

Operator Overloading; String and Array Objects
- Fundamentals of Operator Overloading
- Restrictions on Operator Overloading
- Operator Functions as Class Members vs. Global Functions
- Overloading Stream Insertion and Stream Extraction Operators
- Overloading Unary Operators
- Overloading Binary Operators
- Case Study: Array Class
- Converting `between` Types
- Case Study: `String` Class
- Overloading ++ and `--`
- Case Study: A `Date` Class
- Standard Library Class string
- `explicit` Constructors

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *C++ How to Program, 6/e* (ISBN: 0136152503)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194 Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com (Pearson GSA Contract # GS14F8023A)

C++200—C++ and Object Oriented Programming

Overview
An intensive, 5-day, lecture-and-lab C++ and object-oriented programming course for practicing C (or other high-level language) programmers. C++ offers three major items beyond C, and each is discussed in reasonable technical detail in this course—enhancements to C that improve the process of writing programs, features for data abstraction and encapsulation that enable programmers to create new types, and capabilities for object-oriented programming with inheritance and polymorphism. The course clearly explains the C++ programming language and contains detailed walkthroughs of many C++ programs. Students become proficient in C++, understand the object-oriented paradigm, and participate in challenging hands-on laboratory assignments. Solutions are provided for the laboratory exercises. 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- Programming in C or another high-level language, or

Introduction
Review of C++ syntax and concepts presented in the following chapters of *C++ How to Program, 6/e*.
- Chapter 2—Introduction to C++ Programming
- Chapter 3—Introduction to Classes and Objects
- Chapter 4—Control Statements: Part 1
- Chapter 5—Control Statements: Part 2
- Chapter 6—Functions
- Chapter 7—Arrays
- Chapter 8—Pointers and Pointer-Based Strings

Classes: A Deeper Look, Part 1
- `Time` Class Case Study
- Class Scope and Accessing Class Members
- Separating Interface from Implementation
- Access Functions and Utility Functions
- `Time` Class Case Study: Constructors with Default Arguments
- Destructors
- When Constructors and Destructors Are Called
- `Time` Class Case Study: A Subtle Trap—Returning a Reference to a private Data Member
- Default Memberwise Assignment

Classes: A Deeper Look, Part 2
- `const` (Constant) Objects and const Member Functions
- Composition: Objects as Members of Classes
- `friend` Functions and friend Classes
- Using the `this` Pointer
- Dynamic Memory Management with Operators new and delete
- `static` Class Members
- Data Abstraction and Information Hiding
- Proxy Classes

Operator Overloading; String and Array Objects
- Fundamentals of Operator Overloading
- Restrictions on Operator Overloading
- Operator Functions as Class Members vs. Global Functions

- Overloading Stream Insertion and Stream Extraction Operators
- Overloading Unary Operators
- Overloading Binary Operators
- Case Study: Array Class
- Converting `between` Types
- Case Study: `String` Class
- Overloading ++ and `--`
- Case Study: A `Date` Class
- Standard Library Class string
- `explicit` Constructors

Object-Oriented Programming: Inheritance
- Base Classes and Derived Classes
- `protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class Without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using protected Data
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using private Data
- Constructors and Destructors in Derived Classes
- `public`, `protected` and `private` Inheritance

Object-Oriented Programming: Polymorphism
- Polymorphism Examples
- Relationships Among Objects in an Inheritance Hierarchy
- Invoking Base-Class Functions from Derived-Class Objects
- Aiming Derived-Class Pointers at Base-Class Objects
- Derived-Class Member-Function Calls via Base-Class Pointers
- Virtual Functions
- Summary of the Allowed Assignments Between Base-Class and Derived-Class Objects and Pointers
- Type Fields and `switch` Statements
- Abstract Classes and Pure `virtual` Functions
- Case Study: Payroll System Using Polymorphism
- Demonstrating Polymorphic Processing
- (Optional) Polymorphism, Virtual Functions and Dynamic Binding "Under the Hood"
- Case Study: Payroll System Using Polymorphism and Runtime Type Information with Downcasting, `dynamic_cast`, `typeid` and `type_info`
- Virtual Destructors

Templates
- Function Templates
- Overloading Function Templates
- Class Templates
- Nontype Parameters and Default Types for Class Templates

Exception Handling
- Exception-Handling Overview
- Example: Handling an Attempt to Divide by Zero
- When to Use Exception Handling

- Rethrowing an Exception
- Exception Specifications
- Processing Unexpected Exceptions
- Stack Unwinding
- Constructors, Destructors and Exception Handling
- Exceptions and Inheritance
- Processing new Failures
- Class auto_ptr and Dynamic Memory Allocation
- Standard Library Exception Hierarchy

File Processing

- Data Hierarchy
- Files and Streams
- Creating a Sequential File
- Reading Data from a Sequential File
- Updating Sequential Files
- Random-Access Files
- Creating a Random-Access File
- Writing Data Randomly to a Random-Access File
- Reading from a Random-Access File Sequentially
- Case Study: A Transaction-Processing Program
- Overview of Object Serialization

Standard Template Library (STL)

- Introduction to the Standard Template Library (STL)
- Introduction to Containers
- Introduction to Iterators
- Introduction to Algorithms
- Sequence Containers
  - vector Sequence Container
  - list Sequence Container
  - deque Sequence Container
- Associative Containers
  - multiset Associative Container
  - set Associative Container
  - multimap Associative Container
  - map Associative Container
- Container Adapters
  - stack Adapter
  - queue Adapter
  - priority_queue Adapter
- Algorithms
  - fill, fill_n, generate and generate_n
  - equal, mismatch and lexicographical_compare
  - remove, remove_if, remove_copy and remove_copy_if
  - Basic Searching and Sorting Algorithms
  - copy_backward, merge, unique and reverse
  - Set Operations
- Function Objects

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study
An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

Price

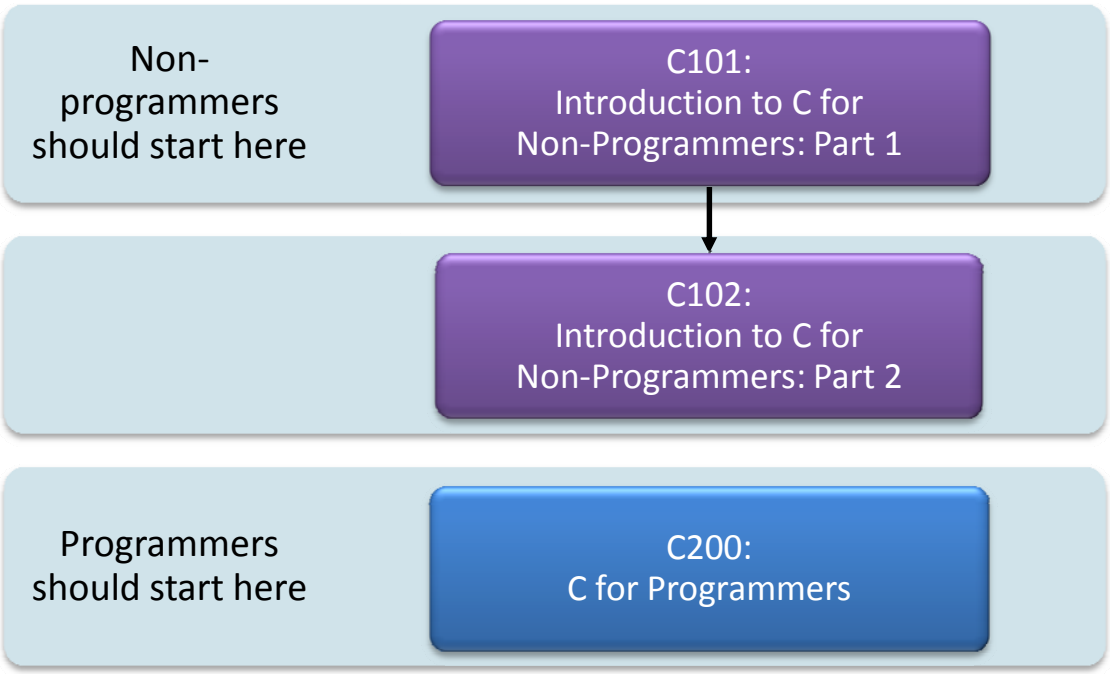
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *C++ How to Program, 6/e* (ISBN: 0136152503)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

C Programming Curriculum

Overview

C is still one of the most popular programming languages in use today. It is the language of choice for high-performance applications.

Deitel & Associates, Inc. is the world's leading programming language textbook/professional book authors and our courses are based on our best-selling books.



C101—Introduction to C for Non-Programmers: Part 1

Overview

This 5-day, carefully paced, hands-on course for novices teaches fundamental C concepts such as algorithmic thinking, problem solving, control structures (`if`, `if…else`, `switch`, `while`, `do…while`, `for`), data types, operators, input/output, functions (user-defined and library) and arrays. The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. After taking this course, students will be prepared to take C102—*Introduction to C for Non-Programmers: Part 2*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Our introductory classes emphasize achieving program clarity through the proven software-development techniques. Nonprogrammers learn skills that underlie good programming through dozens of complete, working programs—we call this the live-code approach. This hands-on approach rapidly builds the confidence of new programmers, because all programming concepts are presented in the context in which they will be used.

Prerequisites

- General familiarity with your operating system environment
- Ability to create and edit text files

Introduction to C Programming

- A Simple C Program: Printing a Line of Text
- Another Simple C Program: Adding Two Integers
- Memory Concepts
- Arithmetic in C
- Decision Making: Equality and Relational Operators

Structured Program Development in C

- Algorithms
- Pseudocode
- Control Structures
- The if Selection Statement
- The if…else Selection Statement
- The while Repetition Statement
- Formulating Algorithms Case Study 1: Counter-Controlled Repetition
- Formulating Algorithms with Top-Down, Stepwise Refinement
- Case Study 2: Sentinel-Controlled Repetition
    - Formulating Algorithms with Top-Down, Stepwise Refinement
- Case Study 3: Nested Control Structures
    - Assignment Operators
    - Increment and Decrement Operators

C Program Control

- Repetition Essentials
- Counter-Controlled Repetition
- `for` Repetition Statement
- `for` Statement: Notes and Observations
- Examples Using the `for` Statement
- `switch` Multiple-Selection Statement
- `do…while` Repetition Statement
- `break` and `continue` Statements

- Logical Operators
- Confusing Equality (==) and Assignment (=) Operators
- Structured Programming Summary

C Functions

- Program Modules in C
- Math Library Functions
- Functions
- Function Definitions
- Function Prototypes
- Function Call Stack and Activation Records
- Headers
- Calling Functions: Call-by-Value and Call-by-Reference
- Random Number Generation
- Example: A Game of Chance
- Storage Classes
- Scope Rules
- Recursion
- Example Using Recursion: Fibonacci Series
- Recursion vs. Iteration

C Arrays

- Arrays
- Defining Arrays
- Array Examples
- Passing Arrays to Functions
- Sorting Arrays
- Case Study: Computing Mean, Median and Mode Using Arrays
- Searching Arrays
- Multiple-Subscripted Arrays

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *C How to Program, 5/e* (ISBN: 0132404168)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

C102—Introduction to C for Non-Programmers: Part 2

Overview

This 5-day, lecture-and-lab course includes pointers, pointer arithmetic, function pointers, strings, optimizing memory use, improving execution performance, formatted I/O, structures, unions, enumerations, bit manipulation and file processing. The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- C101—*Introduction to C for Non-Programmers: Part 1*, or equivalent programming experience

Introduction

Review of C syntax and concepts presented in the following chapters of *C How to Program, 5/e* as part of the course *Introduction to C and C++: Part 1 (for Nonprogrammers)*.

- Chapter 2—Introduction to C Programming
- Chapter 3—Structured Program Development in C
- Chapter 4—C Program Control
- Chapter 5—C Functions
- Chapter 6—C Arrays

C Pointers

- Pointer Variable Definitions and Initialization
- Pointer Operators
- Passing Arguments to Functions by Reference
- Using the const Qualifier with Pointers
- Bubble Sort Using Call-by-Reference
- sizeof Operator
- Pointer Expressions and Pointer Arithmetic
- Relationship between Pointers and Arrays
- Arrays of Pointers
- Case Study: Card Shuffling and Dealing Simulation
- Pointers to Functions

C Characters and Strings

- Fundamentals of Strings and Characters
- Character-Handling Library
- String-Conversion Functions
- Standard Input/Output Library Functions
- String-Manipulation Functions of the String-Handling Library
- Comparison Functions of the String-Handling Library
- Search Functions of the String-Handling Library
- Memory Functions of the String-Handling Library
- Other Functions of the String-Handling Library

C Formatted Input/Output

- Streams
- Formatting Output with printf
- Printing Integers
- Printing Floating-Point Numbers
- Printing Strings and Characters

- Other Conversion Specifiers
- Printing with Field Widths and Precision
- Using Flags in the printf Format Control String
- Printing Literals and Escape Sequences
- Reading Formatted Input with scanf

C Structures, Unions, Bit Manipulations and Enumerations

- Structure Definitions
- Initializing Structures
- Accessing Members of Structures
- Using Structures with Functions
- typedef
- Example: High-Performance Card Shuffling and Dealing Simulation
- Unions
- Bitwise Operators
- Bit Fields
- Enumeration Constants

C File Processing

- Data Hierarchy
- Files and Streams
- Creating a Sequential-Access File
- Reading Data from a Sequential-Access File
- Random-Access Files
- Creating a Random-Access File
- Writing Data Randomly to a Random-Access File
- Reading Data from a Random-Access File
- Case Study: Transaction-Processing Program

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *C How to Program, 5/e* (ISBN: 0132404168)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

C200—C for Programmers

Overview

This 5-day, lecture-and-lab course for programmers who don't know C, teaches fundamental C concepts such as control structures (`if`, `if…else`, `switch`, `while`, `do…while`, `for`), data types, operators, input/output, functions (user-defined and library), arrays, pointers, pointer arithmetic, function pointers, strings, optimizing memory use, improving execution performance, formatted I/O, structures, unions, enumerations, bit manipulation and file processing. The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. Approximately 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- Programming experience in a high-level programming language

Introduction to C Programming

- A Simple C Program: Printing a Line of Text
- Another Simple C Program: Adding Two Integers
- Memory Concepts
- Arithmetic in C
- Decision Making: Equality and Relational Operators

Structured Program Development in C

- Algorithms
- Pseudocode
- Control Structures
- The if Selection Statement
- The if…else Selection Statement
- The while Repetition Statement
- Formulating Algorithms Case Study 1: Counter-Controlled Repetition
- Formulating Algorithms with Top-Down, Stepwise Refinement
- Case Study 2: Sentinel-Controlled Repetition
    - Formulating Algorithms with Top-Down, Stepwise Refinement
- Case Study 3: Nested Control Structures
    - Assignment Operators
    - Increment and Decrement Operators

Program Control

- Repetition Essentials
- Counter-Controlled Repetition
- for Repetition Statement
- for Statement: Notes and Observations
- Examples Using the for Statement
- switch Multiple-Selection Statement
- do…while Repetition Statement
- break and continue Statements
- Logical Operators
- Confusing Equality (==) and Assignment (=) Operators
- Structured Programming Summary

Functions

- Program Modules in C
- Math Library Functions
- Functions
- Function Definitions
- Function Prototypes
- Function Call Stack and Activation Records
- Headers
- Calling Functions: Call-by-Value and Call-by-Reference
- Random Number Generation
- Example: A Game of Chance
- Storage Classes
- Scope Rules
- Recursion
- Example Using Recursion: Fibonacci Series
- Recursion vs. Iteration

Arrays

- Arrays
- Defining Arrays
- Array Examples
- Passing Arrays to Functions
- Sorting Arrays
- Case Study: Computing Mean, Median and Mode Using Arrays
- Searching Arrays
- Multiple-Subscripted Arrays

Pointers

- Pointer Variable Definitions and Initialization
- Pointer Operators
- Passing Arguments to Functions by Reference
- Using the const Qualifier with Pointers
- Bubble Sort Using Call-by-Reference
- sizeof Operator
- Pointer Expressions and Pointer Arithmetic
- Relationship between Pointers and Arrays
- Arrays of Pointers
- Case Study: Card Shuffling and Dealing Simulation
- Pointers to Functions

Characters and Strings

- Fundamentals of Strings and Characters
- Character-Handling Library
- String-Conversion Functions
- Standard Input/Output Library Functions
- String-Manipulation Functions of the String-Handling Library
- Comparison Functions of the String-Handling Library
- Search Functions of the String-Handling Library
- Memory Functions of the String-Handling Library
- Other Functions of the String-Handling Library

Formatted Input/Output

- Streams
- Formatting Output with printf
- Printing Integers
- Printing Floating-Point Numbers
- Printing Strings and Characters
- Other Conversion Specifiers
- Printing with Field Widths and Precision
- Using Flags in the printf Format Control String
- Printing Literals and Escape Sequences
- Reading Formatted Input with scanf

Structures, Unions, Bit Manipulations and Enumerations

- Structure Definitions
- Initializing Structures
- Accessing Members of Structures
- Using Structures with Functions
- typedef
- Example: High-Performance Card Shuffling and Dealing Simulation
- Unions
- Bitwise Operators
- Bit Fields
- Enumeration Constants

File Processing

- Data Hierarchy
- Files and Streams
- Creating a Sequential-Access File
- Reading Data from a Sequential-Access File
- Random-Access Files
- Creating a Random-Access File
- Writing Data Randomly to a Random-Access File
- Reading Data from a Random-Access File
- Case Study: Transaction-Processing Program

Price

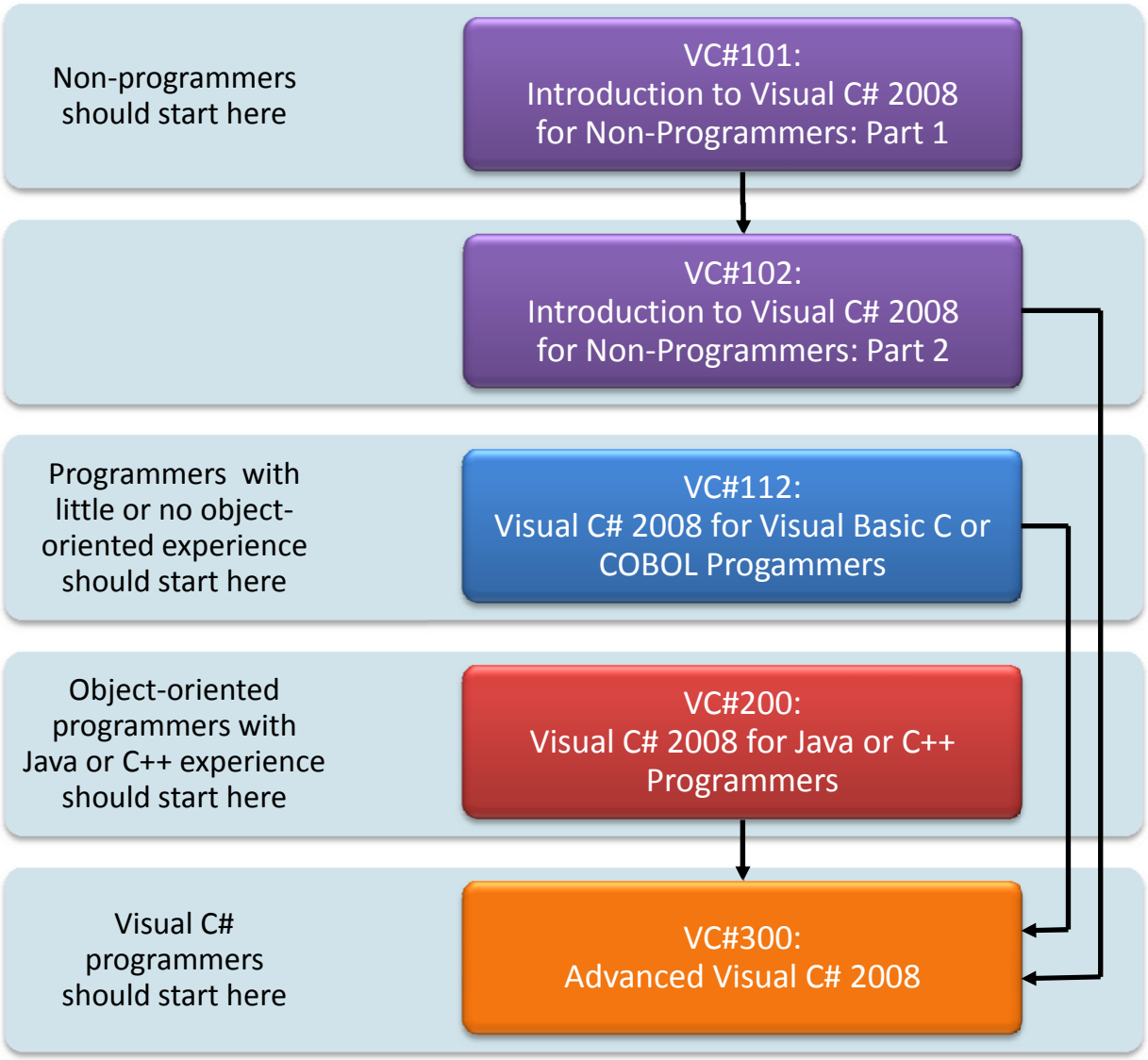
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *C How to Program, 5/e* (ISBN: 0132404168)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Visual C#® 2008 Programming Curriculum

Overview

Visual C# is the up-and-coming programming language that Microsoft designed specifically for the .NET platform. C# has become a language of choice for implementing Windows applications that integrate with the Internet and the Web. These classes help novices become literate in Visual C# programming and help experienced programmers become skilled Visual C# developers. The classes are up-to-the-minute with Microsoft's latest development tools and the .NET 3.5 platform.

Deitel & Associates, Inc. is the world's leading programming language textbook/professional book authors and our courses are based on our best-selling books.



VC#101—Introduction to Visual C# 2008 for Non-Programmers: Part 1

Overview

This introductory, 5-day, lecture-and-lab course teaches people with little or no programming experience how to program with the Visual C# 2008 programming language. Topics include introductory object-oriented programming, algorithmic thinking, problem solving, control statements, primitive types, operators, keyboard input, screen output, methods (user-defined and API), arrays and strings. The course offers extensive laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. After taking this course, students will be prepared to take VC#102—*Introduction to Visual C# 2008 for Non-Programmers: Part 2*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Our introductory classes emphasize achieving program clarity through the proven software-development techniques. Nonprogrammers learn skills that underlie good programming through dozens of complete, working programs—we call this the live-code approach. This hands-on approach rapidly builds the confidence of new programmers, because all programming concepts are presented in the context in which they will be used.

Prerequisites

- Little or no previous programming experience
- General familiarity with your operating system environment
- Ability to create and edit text files

Introduction

- Overview of *Visual C# 2008 How to Program, 3/e*
- Overview of the Visual Studio tools for building C# applications

Introduction to the Visual C# 2008 Express IDE

- Covered in the context of all the examples presented

Introduction to C# Applications

- A Simple C# Application: Displaying a Line of Text
- Creating Your Simple Application in Visual C# Express
- Modifying Your Simple C# Application
- Formatting Text with `Console.Write` and `Console.WriteLine`
- Another C# Application: Adding Integers
- Decision Making: Equality and Relational Operators

Introduction to Classes and Objects

- Classes, Objects, Methods, Properties and Instance Variables
- Declaring a Class with a Method and Instantiating an Object of a Class
- Declaring a Method with a Parameter
- Instance Variables, Properties and Auto-Implemented Properties
- Software Engineering with Properties and set and get Accessors
- Value Types vs. Reference Types
- Initializing Objects with Constructors
- Floating-Point Numbers and Type `decimal`

Control Statements: Part 1

- Algorithms
- Pseudocode

- Control Structures
- `if` Single-Selection Statement
- `if…else` Double-Selection Statement
- `while` Repetition Statement
- Formulating Algorithms: Counter-Controlled Repetition
- Formulating Algorithms: Sentinel-Controlled Repetition
- Formulating Algorithms: Nested Control Statements
- Compound Assignment Operators
- Increment and Decrement Operators
- Simple Types

Control Statements: Part 2
- Essentials of Counter-Controlled Repetition
- `for` Repetition Statement
- Examples Using the `for` Statement
- `do…while` Repetition Statement
- `switch` Multiple-Selection Statement
- `break` and continue Statements
- Logical Operators

Methods: A Deeper Look
- `static` Methods, `static` Variables and Class `Math`
- Declaring Methods with Multiple Parameters
- Notes on Declaring and Using Methods
- Method Call Stack and Activation Records
- Argument Promotion and Casting
- The Framework Class Library
- Case Study: Random-Number Generation
- Scaling and Shifting Random Numbers
- Random-Number Repeatability for Testing and Debugging
- Case Study: A Game of Chance (Introducing Enumerations)
- Scope of Declarations
- Method Overloading
- Passing Arguments: Pass-by-Value vs. Pass-by-Reference

Arrays
- Declaring and Creating Arrays
- Examples Using Arrays
- Case Study: Card Shuffling and Dealing Simulation
- `foreach` Statement
- Passing Arrays and Array Elements to Methods
- Passing Arrays by Value and by Reference
- Case Study: Class `GradeBook` Using an Array to Store Grades
- Multidimensional Arrays
- Case Study: Class `GradeBook` Using a Rectangular Array
- Variable-Length Argument Lists
- Using Command-Line Arguments

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C# 2008 How to Program, 3/e* (ISBN: 013605322X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VC#102—Introduction to Visual C# 2008 for Non-Programmers: Part 2

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with Visual C# 2008. Key topics include the Visual C# 2008 Express IDE (or Visual Studio 2008), classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, creating class libraries, Language Integrated Query (LINQ), an introduction to generic collections, event-driven programming, graphical user interfaces (GUIs) with Windows Forms and exception handling. Students create C# applications, understand C# object-oriented programming, learn to use various .NET Framework Class Library classes and participate in extensive laboratory assignments for hands on experience. After taking this course and gaining some practical Visual C# programming experience, students will be prepared to take VC#300—*Advanced Visual C# 2008*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- VC#101—*Introduction to Visual C# for Non-Programmers: Part 1*, or
- Ability to program in a high-level language.

Introduction

Review of C# syntax and concepts presented in the following chapters of *Visual C# 2008 How to Program, 3/e*:

- Chapter 2—Introduction to the Visual C# 2008 Express IDE
- Chapter 3—Introduction to C# Applications
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Methods: A Deeper Look
- Chapter 8—Arrays

Introduction to LINQ and Generic Collections

- Querying an Array Using LINQ
- Introduction to Collections
- Querying a Generic Collection Using LINQ
- Deitel LINQ Resource Center

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Controlling Access to Members
- Referring to the Current Object's Members with the `this` Reference
- Indexers
- `Time` Class Case Study: Overloaded Constructors
- Default and Parameterless Constructors
- Composition
- Garbage Collection and Destructors
- `static` Class Members
- `readonly` Instance Variables
- `Time` Class Case Study: Creating Class Libraries
- Class View and Object Browser

Object-Oriented Programming: Inheritance

- Base Classes and Derived Classes
- `protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using protected Instance Variables
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using private Instance Variables
- Class `object`

Polymorphism, Interfaces & Operator Overloading

- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Using Polymorphism
- Operator `is` and Downcasting
- Summary of the Allowed Assignments Between Base Class and Derived Class Variables
- `sealed` Methods and Classes
- Case Study: Creating and Using Interfaces
- Common Interfaces of the .NET Framework Class Library
- Operator Overloading

Exception Handling

- Exception Handling Overview
- Example: Divide by Zero Without Exception Handling
- Handling `DivideByZeroExceptions` and `FormatExceptions`
- Enclosing Code in a `try` Block
- Catching Exceptions
- Uncaught Exceptions
- Termination Model of Exception Handling
- Flow of Control When Exceptions Occur
- .NET Exception Hierarchy
- Determining Which Exceptions a Method Throws
- `finally` Block
- Exception Properties
- User-Defined Exception Classes

Graphical User Interfaces with Windows Forms

- Windows Forms
- Event Handling
- A Simple Event-Driven GUI
- Another Look at the Visual Studio Generated Code
- Delegates and the Event-Handling Mechanism
- Other Ways to Create Event Handlers
- Locating Event Information
- Control Properties and Layout
- Labels, TextBoxes and Buttons

- GroupBoxes and Panels
- CheckBoxes and RadioButtons
- PictureBoxes
- ToolTips
- NumericUpDown Control
- Mouse-Event Handling

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C# 2008 How to Program, 3/e* (ISBN: 013605322X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VC#112—Visual C# 2008 for Visual Basic, C or COBOL Programmers

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with Visual C# 2008. Key topics include the Visual C# 2008 Express IDE (or Visual Studio 2008), classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, creating class libraries, Language Integrated Query (LINQ), an introduction to generic collections, event-driven programming, graphical user interfaces (GUIs) with Windows Forms and exception handling. Students create C# applications, understand C# object-oriented programming, learn to use various .NET Framework Class Library classes and participate in extensive laboratory assignments for hands on experience. After taking this course, students will be prepared to take VC#300—*Advanced Visual C# 2008*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- VC#101—*Introduction to Visual C# for Non-Programmers: Part 1*, or
- Ability to program in Visual Basic, C or COBOL.

Introduction

Review of C# syntax and concepts presented in the following chapters of *Visual C# 2008 How to Program, 3/e*:

- Chapter 2—Introduction to the Visual C# 2008 Express IDE
- Chapter 3—Introduction to C# Applications
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Methods: A Deeper Look
- Chapter 8—Arrays

Introduction to LINQ and Generic Collections

- Querying an Array Using LINQ
- Introduction to Collections
- Querying a Generic Collection Using LINQ
- Deitel LINQ Resource Center

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Controlling Access to Members
- Referring to the Current Object's Members with the `this` Reference
- Indexers
- `Time` Class Case Study: Overloaded Constructors
- Default and Parameterless Constructors
- Composition
- Garbage Collection and Destructors
- `static` Class Members
- `readonly` Instance Variables
- `Time` Class Case Study: Creating Class Libraries
- Class View and Object Browser

Object-Oriented Programming: Inheritance

- Base Classes and Derived Classes

- `protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using protected Instance Variables
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using private Instance Variables
- Class `object`

Polymorphism, Interfaces & Operator Overloading

- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Using Polymorphism
- Operator `is` and Downcasting
- Summary of the Allowed Assignments Between Base Class and Derived Class Variables
- `sealed` Methods and Classes
- Case Study: Creating and Using Interfaces
- Common Interfaces of the .NET Framework Class Library
- Operator Overloading

Exception Handling

- Exception Handling Overview
- Example: Divide by Zero Without Exception Handling
- Handling `DivideByZeroException`s and `FormatException`s
- Enclosing Code in a `try` Block
- Catching Exceptions
- Uncaught Exceptions
- Termination Model of Exception Handling
- Flow of Control When Exceptions Occur
- .NET Exception Hierarchy
- Determining Which Exceptions a Method Throws
- `finally` Block
- Exception Properties
- User-Defined Exception Classes

Graphical User Interfaces with Windows Forms

- Windows Forms
- Event Handling
- A Simple Event-Driven GUI
- Another Look at the Visual Studio Generated Code
- Delegates and the Event-Handling Mechanism
- Other Ways to Create Event Handlers
- Locating Event Information
- Control Properties and Layout
- Labels, TextBoxes and Buttons
- GroupBoxes and Panels
- CheckBoxes and RadioButtons

- PictureBoxes
- ToolTips
- NumericUpDown Control
- Mouse-Event Handling

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C# 2008 How to Program, 3/e* (ISBN: 013605322X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VC#200—Visual C# 2008 for Java or C++ Programmers

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with Visual C# 2008. Key topics include the Visual C# 2008 Express IDE (or Visual Studio 2008), classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, creating class libraries, Language Integrated Query (LINQ), generic collections, event-driven programming, Windows Forms graphical user interfaces (GUIs), Windows Presentation Foundation (WPF) GUI and graphics, exception handling, file processing and generics. The object-oriented concepts in this course are presented with a focus on the differences between C# and other programming languages. Students create Visual C# applications, understand C# object-oriented programming, learn to use various .NET Framework Class Library classes and participate in extensive laboratory assignments for hands on experience. After taking this course, students will be prepared to take VC#300—*Advanced Visual C# 2008*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- Object-oriented programming in a language such as C++ or Java.

Introduction

Review of C# syntax and concepts presented in the following chapters of *Visual C# 2008 How to Program, 3/e*:

- Chapter 2—Introduction to the Visual C# 2008 Express IDE
- Chapter 3—Introduction to C# Applications
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Methods: A Deeper Look
- Chapter 8—Arrays

Introduction to LINQ and Generic Collections

- Querying an Array Using LINQ
- Introduction to Collections
- Querying a Generic Collection Using LINQ
- Deitel LINQ Resource Center

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Controlling Access to Members
- Referring to the Current Object's Members with the `this` Reference
- Indexers
- `Time` Class Case Study: Overloaded Constructors
- Default and Parameterless Constructors
- Composition
- Garbage Collection and Destructors
- `static` Class Members
- `readonly` Instance Variables
- `Time` Class Case Study: Creating Class Libraries
- Class View and Object Browser

Object-Oriented Programming: Inheritance

- Base Classes and Derived Classes
- `protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using protected Instance Variables
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using private Instance Variables
- Class `object`

Polymorphism, Interfaces & Operator Overloading

- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Using Polymorphism
- Operator `is` and Downcasting
- Summary of the Allowed Assignments Between Base Class and Derived Class Variables
- `sealed` Methods and Classes
- Case Study: Creating and Using Interfaces
- Common Interfaces of the .NET Framework Class Library
- Operator Overloading

Exception Handling

- Exception Handling Overview
- Example: Divide by Zero Without Exception Handling
- Handling `DivideByZeroExceptions` and `FormatExceptions`
- Enclosing Code in a `try` Block
- Catching Exceptions
- Uncaught Exceptions
- Termination Model of Exception Handling
- Flow of Control When Exceptions Occur
- .NET Exception Hierarchy
- Determining Which Exceptions a Method Throws
- `finally` Block
- Exception Properties
- User-Defined Exception Classes

Graphical User Interfaces with Windows Forms

- Windows Forms
- Event Handling
- A Simple Event-Driven GUI
- Another Look at the Visual Studio Generated Code
- Delegates and the Event-Handling Mechanism
- Other Ways to Create Event Handlers
- Locating Event Information
- Control Properties and Layout
- Labels, TextBoxes and Buttons

- GroupBoxes and Panels
- CheckBoxes and RadioButtons
- PictureBoxes
- ToolTips
- NumericUpDown Control
- Mouse-Event Handling

GUI with Windows Presentation Foundation
- Windows Presentation Foundation (WPF)
- XML Basics
- Structuring Data
- XML Namespaces
- Declarative GUI Programming Using XAML
- Creating a WPF Application in Visual Basic Express
- Laying Out Controls
- General Layout Principles
- Layout in Action
- Event Handling
- Commands and Common Application Tasks
- WPF GUI Customization
- Using Styles to Change the Appearance of Controls
- Customizing Windows
- Defining a Control's Appearance with Control Templates
- Data-Driven GUIs with Data Binding

WPF Graphics and Multimedia
- Controlling Fonts
- Basic Shapes
- Polygons and Polylines
- Brushes
- Transforms
- WPF Customization: A Television GUI
- Animations

Files and Streams
- Classes `File` and `Directory`
- Creating a Sequential-Access Text File
- Reading Data from a Sequential-Access Text File
- Serialization
- Creating a Sequential-Access File Using Object Serialization
- Reading and Deserializing Data from a Sequential-Access File

Generics
- Motivation for Generic Methods
- Generic Method Implementation
- Type Constraints
- Overloading Generic Methods
- Generic Classes
- Notes on Generics and Inheritance

Collections

- Introduction
- Collections Overview
- Class `Array` and `Enumerators`
- Generic Class `SortedDictionary`
- Generic Class `LinkedList`

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study

An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C# 2008 How to Program, 3/e* (ISBN: 013605322X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VC#300—Advanced Visual C# 2008

Prerequisites
This 5-day, lecture-and-lab course presents enterprise and web-application development techniques in Visual C# 2008. Key topics include Language Integrated Query (LINQ), XML, LINQ to XML, Windows Presentation Foundation (WPF), Manipulating databases with LINQ to SQL, SQL Server Express, ASP.NET 3.5, ASP.NET AJAX, building web services with Windows Communication Foundation (WCF), consuming WCF web services and building Rich Internet Applications (RIAs) with Silverlight. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- VC#200—*Visual C# 2008 for Java or C++ Programmers*, or
- VC#112— *Visual C# 2008 for Visual Basic, C or COBOL Programmers*, or
- VC#102—*Introduction to Visual C# 2008 for Non-Programmers: Part 2*, plus practical Visual C# programming experience, or
- Equivalent Visual C# 2008 programming experience

Introduction
- Review of Chapter 9—Introduction to LINQ and Generic Collections
- Review of Chapter 16—GUI with Windows Presentation Foundation

XML and LINQ to XML
- W3C XML Schema Documents
- Extensible Stylesheet Language and XSL Transformations
- LINQ to XML: Document Object Model (DOM)
- LINQ to XML: XML Axis Properties
- LINQ to XML: XML Literals and Embedded Expressions
- XSLT with Class XslCompiledTransform

Database, SQL and LINQ to SQL
- Relational Databases
- Relational Database Overview: Books Database
- Overview Structured Query Language (SQL)
- LINQ to SQL (SQL in this case means Microsoft's SQL Server DBMS)
- Creating LINQ to SQL Classes
- Creating Data Bindings
- More Complex LINQ Queries and Data Binding
- Retrieving Data from Multiple Tables with LINQ
- Creating a Master/Detail View Application
- Programming with LINQ to SQL: Address-Book Case Study

ASP.NET 3.5 and ASP.NET AJAX
- Simple HTTP Transactions
- Multitier Application Architecture
- Creating and Running a Simple Web-Form Example
- Examining an ASPX File
- Examining a Code-Behind File
- Relationship Between an ASPX File and a Code-Behind File
- How the Code in an ASP.NET Web Page Executes
- Examining the XHTML Generated by an ASP.NET Application

- Building an ASP.NET Web Application
- Web Controls
    - Text and Graphics Controls
    - AdRotator Control
    - Validation Controls
- Session Tracking
    - Cookies
    - Session Tracking with `HttpSessionState`
- Case Study: Guestbook Application—Connecting to a Database in ASP.NET
- Case Study: Secure Books Database Application
- ASP.NET AJAX
    - Traditional Web Applications
    - Ajax Web Applications
    - Examining an ASP.NET AJAX Application
- New ASP.NET 3.5 Data Controls

Windows Communication Foundation (WCF) Web Services

- WCF Services Basics
- Simple Object Access Protocol (SOAP)
- Representational State Transfer (REST)
- JavaScript Object Notation (JSON)
- Publishing and Consuming SOAP-Based WCF Web Services
    - Creating a WCF Web Service
    - Code for the `WelcomeSOAPXMLService`
    - Building a SOAP WCF Web Service
    - Deploying the `WelcomeSOAPXMLService`
    - Creating a Client to Consume the `WelcomeSOAPXMLService`
    - Consuming the `WelcomeSOAPXMLService`
- Publishing and Consuming REST-Based XML Web Services
    - Creating a REST-Based XML WCF Web Service
    - Consuming a REST-Based XML WCF Web Service
- Publishing and Consuming REST-Based JSON Web Services
    - Creating a REST-Based JSON WCF Web Service
    - Consuming a REST-Based JSON WCF Web Service
- Blackjack Web Service
    - Using Session Tracking in a SOAP-Based WCF Web Service
    - Creating a Blackjack Web Service
    - Consuming the Blackjack Web Service
- Airline Reservation Web Service
    - Database Access and Invoking a Service from ASP.NET
- Equation Generator: Returning User-Defined Types
    - Creating the REST-Based XML EquationGenerator Web Service
    - Consuming the REST-Based XML EquationGenerator Web Service
    - Creating the REST-Based JSON WCF EquationGenerator Web Service
    - Consuming the REST-Based JSON WCF EquationGenerator Web Service

Silverlight, Rich Internet Applications and Multimedia

- Platform Overview
- Silverlight Demos
- Silverlight Runtime and Tools Installation
- Building a Silverlight `WeatherViewer` Application

- GUI Layout
- Obtaining and Displaying Weather Forecast Data
- Custom Controls
- Animations and the `FlickrViewer`
- Images and Deep Zoom
    - Getting Started With Deep Zoom Composer
    - Creating a Silverlight Deep Zoom Application
- Audio and Video
- Isolated Storage

Price

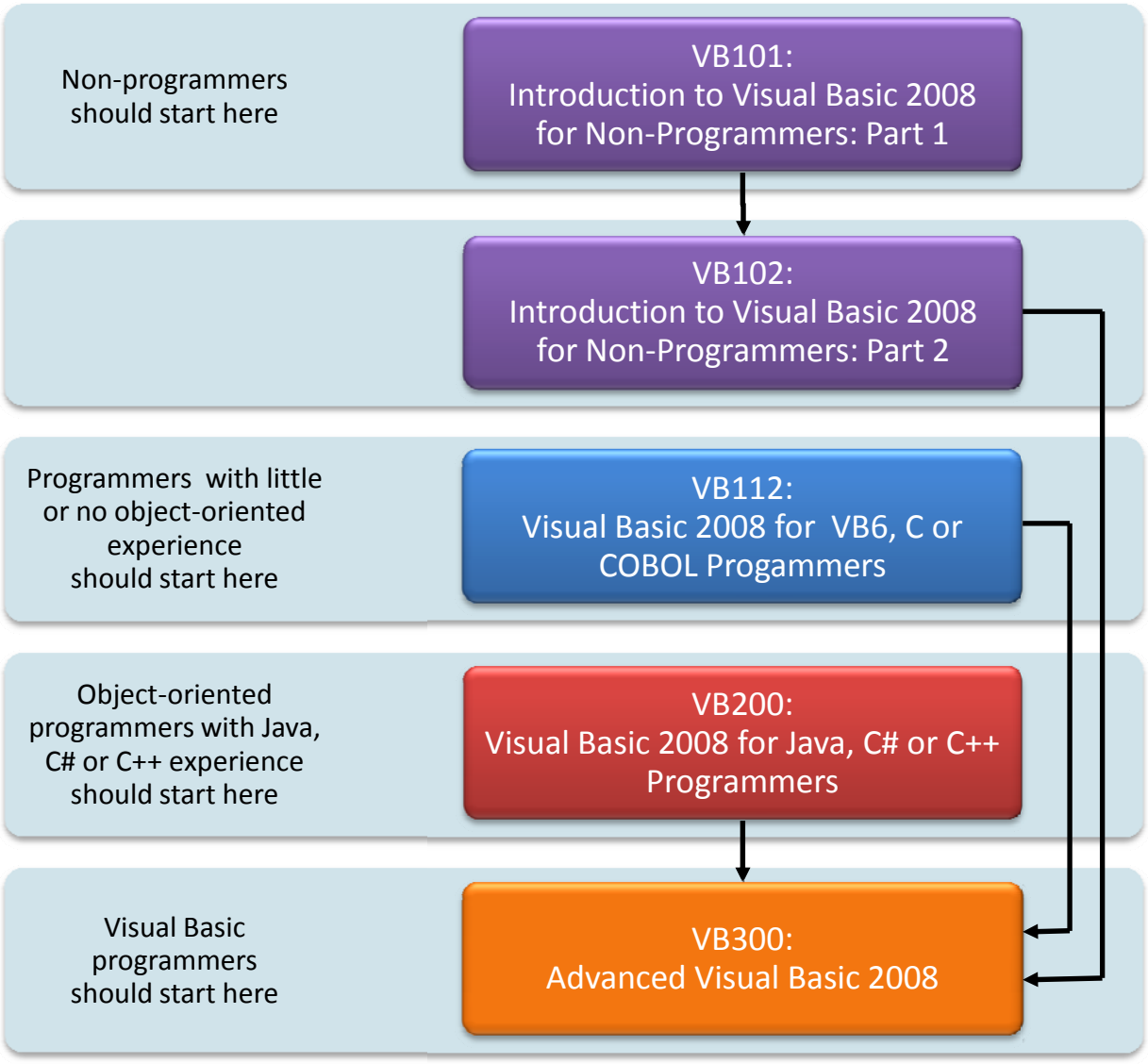
- $13,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C# 2008 How to Program, 3/e* (ISBN: 013605322X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Visual Basic® 2008 Programming Curriculum

Overview

Visual Basic is the second most popular programming language (behind Java) worldwide and is the language of choice for implementing Windows applications that integrate with the Internet and the Web. These classes help novices become literate in Visual Basic programming and help experienced programmers become skilled Visual Basic developers. The classes are up-to-the-minute with Microsoft's latest development tools and the .NET 3.5 platform.

Deitel & Associates, Inc. is the world's leading programming language textbook/professional book authors and our courses are based on our best-selling books.



VB101—Introduction to Visual Basic 2008 for Non Programmers: Part 1

Overview
This introductory 5-day, lecture and laboratory course teaches people with little or no programming experience how to program with the Visual Basic 2008 programming language. Topics include introductory object-oriented programming, algorithmic thinking, problem solving, control statements, primitive types, operators, keyboard input, screen output, methods (user-defined and API), arrays and strings. The course offers extensive laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. After taking this course, students will be prepared to take VB201—*Introduction to Visual Basic 2008 for Non-Programmers: Part 2*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Our introductory classes emphasize achieving program clarity through the proven software-development techniques. Nonprogrammers learn skills that underlie good programming through dozens of complete, working programs—we call this the live-code approach. This hands-on approach rapidly builds the confidence of new programmers, because all programming concepts are presented in the context in which they will be used.

Prerequisites
- Little or no previous programming experience
- General familiarity with your operating system environment
- Ability to create and edit text files

Introduction
- Overview of *Visual Basic 2008 How to Program*
- Overview of the Visual Studio tools for building Visual Basic applications

Introduction to the Visual Basic Express 2008 IDE
- Covered in the context of all the examples presented

Introduction to Visual Basic Programming
- Displaying a Line of Text
- Creating Your First Program in Visual Basic Express
- Displaying a Single Line of Text with Multiple Statements
- Adding Integers
- Decision Making: Equality and Relational Operators
- Using a Message Dialog to Display a Message

Introduction to Classes and Objects
- Classes, Objects, Methods and Instance Variables
- Declaring a Class with a Method and Instantiating an Object of a Class
- Declaring a Method with a Parameter
- Instance Variables and Properties
- Value Types and Reference Types
- Initializing Objects with Constructors
- Validating Data with `Set` Accessors in Properties

Control Statements: Part 1
- Algorithms
- Pseudocode

- Control Structures
- `If`…`Then` Selection Statement
- `If`…`Then`…`Else` Selection Statement
- `While` Repetition Statement
- `Do While`…`Loop` Repetition Statement
- `Do Until`…`Loop` Repetition Statement
- Compound Assignment Operators
- Formulating Algorithms: Counter-Controlled Repetition
- Formulating Algorithms: Sentinel-Controlled Repetition
- Formulating Algorithms: Nested Control Statements
- Formulating Algorithms: Nested Repetition Statements
- Visual Basic Programming in a Windows Forms Application

Control Statements: Part 2
- Essentials of Counter-Controlled Repetition
- `For`…`Next` Repetition Statement
- Examples Using the `For`…`Next` Statement
- `GradeBook` Case Study: `Select`…`Case` Multiple-Selection Statement
- `Do`…`Loop While` Repetition Statement
- `Do`…`Loop Until` Repetition Statement
- Using `Exit` in Repetition Statements
- Using `Continue` in Repetition Statements
- Logical Operators

Methods: A Deeper Look
- Modules, Classes and Methods
- Subroutines: Methods That Do Not Return a Value
- Functions: Methods That Return a Value
- `Shared` Methods and Class `Math`
- `GradeBook` Case Study: Declaring Methods with Multiple Parameters
- Notes on Declaring and Using Methods
- Method Call Stack and Activation Records
- Implicit Argument Conversions
- `Option Strict` and Data-Type Conversions
- Value Types and Reference Types
- Framework Class Library Namespaces
- Passing Arguments: Pass-by-Value vs. Pass-by-Reference
- Scope of Declarations
- Case Study: Random-Number Generation
- Case Study: A Game of Chance
- Method Overloading
- Optional Parameters

Arrays
- Declaring and Allocating Arrays
- Examples Using Arrays
- Allocating an Array
- Initializing the Values in an Array
- Summing the Elements of an Array
- Using Arrays to Analyze Survey Results

- Using Bar Charts to Display Array Data Graphically
- Using the Elements of an Array as Counters
- Case Study: Card Shuffling and Dealing Simulation
- Passing an Array to a Method
- `For Each…Next` Repetition Statement
- `GradeBook` Case Study: Using an Array to Store Grades
- Sorting an Array with Method `Sort` of Class `Array`
- Searching Arrays
- Rectangular Arrays
- `GradeBook` Case Study: Using a Rectangular Array
- Variable-Length Parameter Lists
- Jagged Arrays
- Changing the Size of an Array at Execution Time: Using the `ReDim` Statement
- Passing Arrays: By`Val` vs. By`Ref`

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual Basic 2008 How to Program* (ISBN: 013605305X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VB102—Introduction to Visual Basic 2008 for Non-Programmers: Part 2

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with Visual Basic 2008. Key topics include the Visual Basic 2008 Express IDE (or Visual Studio 2008), classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, creating class libraries, Language Integrated Query (LINQ), an introduction to generic collections, event-driven programming, graphical user interfaces (GUIs) with Windows Forms and exception handling. Students create Visual Basic applications, understand Visual Basic object-oriented programming, learn to use various .NET Framework Class Library classes and participate in extensive laboratory assignments for hands on experience. After taking this course and gaining practical Visual Basic 2008 programming experience, students will be prepared to take VB#300—*Advanced Visual Basic 2008*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- VB101—*Introduction to Visual Basic 2008 for Non-Programmers: Part 1*

Introduction

Review of Visual Basic syntax and concepts presented in the following chapters of *Visual Basic 2008 How to Program*:

- Chapter 2—Introduction to the Visual Basic Express 2008 IDE
- Chapter 3—Introduction to Visual Basic Programming
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Methods: A Deeper Look
- Chapter 8—Arrays

Introduction to LINQ and Generic Collections

- Querying an Array Using LINQ
- Introduction to Collections
- Querying a Generic Collection Using LINQ
- Deitel LINQ Resource Center

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Class Scope
- Object Initializers
- Default and Parameterless Constructors
- `Time` Class Case Study: Overloaded Constructors
- Partial Classes
- Composition
- Using `Me` to Access the Current Object
- Garbage Collection
- `Shared` Class Members
- `Const` and `ReadOnly` Members
- Object Browser
- `Time` Class Case Study: Creating Class Libraries
- `Time` Class Case Study: Extension Methods
- Delegates

- Lambda Expressions
- Anonymous Types

Object-Oriented Programming: Inheritance
- Base Classes and Derived Classes
- `Protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using Protected Instance Variables
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using Private Instance Variables
- Constructors in Derived Classes
- Software Engineering with Inheritance
- Class `Object`
- `Friend` Members

Object-Oriented Programming: Polymorphism
- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Class Hierarchy Using Polymorphism
- Demonstrating Polymorphic Processing, Expression `TypeOf…Is`, `TryCast` and Downcasting
- Summary of the Allowed Assignments between Base-Class and Derived-Class Variables
- `NotOverridable` Methods and `NotInheritable` Classes
- Case Study: Creating and Using Interfaces
- Common Interfaces of the .NET Framework Class Library

Exception Handling
- Exception-Handling Overview
- Example: Divide by Zero Without Exception Handling
- Example: Handling `DivideByZeroException`s and `FormatException`s
- Enclosing Code in a Try Block
- Catching Exceptions
- Uncaught Exceptions
- Termination Model of Exception Handling
- Flow of Control When Exceptions Occur
- .NET Exception Hierarchy
- Determining Which Exceptions a Method Throws
- `Finally` Block
- Exception Properties
- User-Defined Exception Classes

Graphical User Interfaces with Windows Forms: Part 1
- Windows Forms
- Event Handling
- A Simple Event-Driven GUI
- Another Look at the Visual Studio Generated Code

- Delegates and the Event-Handling Mechanism
- Other Ways to Create Event Handlers
- Locating Event Information
- Control Properties and Layout
- `Labels`, `TextBoxes` and `Buttons`
- `GroupBoxes` and `Panels`
- `CheckBoxes` and `RadioButtons`
- `PictureBoxes`
- `ToolTips`
- `NumericUpDown` Control
- Mouse-Event Handling

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study
An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual Basic 2008 How to Program* (ISBN: 013605305X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VB112—Visual Basic 2008 for VB6, C or COBOL Programmers

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with Visual Basic 2008. Key topics include the Visual Basic 2008 Express IDE (or Visual Studio 2008), classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, creating class libraries, Language Integrated Query (LINQ), an introduction to generic collections, event-driven programming, graphical user interfaces (GUIs) with Windows Forms and exception handling. Students create Visual Basic applications, understand Visual Basic object-oriented programming, learn to use various .NET Framework Class Library classes and participate in extensive laboratory assignments for hands on experience. After taking this course, students will be prepared to take VB#300—*Advanced Visual Basic 2008.* The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- Ability to program in VB6, C or COBOL

Introduction

Review of Visual Basic syntax and concepts presented in the following chapters of *Visual Basic 2008 How to Program*:

- Chapter 2—Introduction to the Visual Basic Express 2008 IDE
- Chapter 3—Introduction to Visual Basic Programming
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Methods: A Deeper Look
- Chapter 8—Arrays

Introduction to LINQ and Generic Collections

- Querying an Array Using LINQ
- Introduction to Collections
- Querying a Generic Collection Using LINQ
- Deitel LINQ Resource Center

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Class Scope
- Object Initializers
- Default and Parameterless Constructors
- `Time` Class Case Study: Overloaded Constructors
- Partial Classes
- Composition
- Using `Me` to Access the Current Object
- Garbage Collection
- `Shared` Class Members
- `Const` and `ReadOnly` Members
- Object Browser
- `Time` Class Case Study: Creating Class Libraries
- `Time` Class Case Study: Extension Methods
- Delegates

- Lambda Expressions
- Anonymous Types

Object-Oriented Programming: Inheritance
- Base Classes and Derived Classes
- `Protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using Protected Instance Variables
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using Private Instance Variables
- Constructors in Derived Classes
- Software Engineering with Inheritance
- Class `Object`
- `Friend` Members

Object-Oriented Programming: Polymorphism
- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Class Hierarchy Using Polymorphism
- Demonstrating Polymorphic Processing, Expression `TypeOf…Is`, `TryCast` and Downcasting
- Summary of the Allowed Assignments between Base-Class and Derived-Class Variables
- `NotOverridable` Methods and `NotInheritable` Classes
- Case Study: Creating and Using Interfaces
- Common Interfaces of the .NET Framework Class Library

Exception Handling
- Exception-Handling Overview
- Example: Divide by Zero Without Exception Handling
- Example: Handling `DivideByZeroException`s and `FormatException`s
- Enclosing Code in a Try Block
- Catching Exceptions
- Uncaught Exceptions
- Termination Model of Exception Handling
- Flow of Control When Exceptions Occur
- .NET Exception Hierarchy
- Determining Which Exceptions a Method Throws
- `Finally` Block
- Exception Properties
- User-Defined Exception Classes

Graphical User Interfaces with Windows Forms: Part 1
- Windows Forms
- Event Handling
- A Simple Event-Driven GUI
- Another Look at the Visual Studio Generated Code

- Delegates and the Event-Handling Mechanism
- Other Ways to Create Event Handlers
- Locating Event Information
- Control Properties and Layout
- `Labels`, `TextBoxes` and `Buttons`
- `GroupBoxes` and `Panels`
- `CheckBoxes` and `RadioButtons`
- `PictureBoxes`
- `ToolTips`
- `NumericUpDown` Control
- Mouse-Event Handling

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study
An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual Basic 2008 How to Program* (ISBN: 013605305X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VB200—Visual Basic 2008 for Java, C# or C++ Programmers

Overview

This 5-day, lecture-and-lab course presents object-oriented programming with Visual Basic 2008. Key topics include the Visual Basic 2008 Express IDE (or Visual Studio 2008), classes, objects, encapsulation, interfaces, inheritance, polymorphism, abstract classes, creating class libraries, Language Integrated Query (LINQ), generic collections, event-driven programming, Windows Forms graphical user interfaces (GUIs), Windows Presentation Foundation (WPF) GUI and graphics, exception handling, file processing and generics. The object-oriented concepts in this course are presented with a focus on the differences between Visual Basic and other programming languages. Students create Visual Basic applications, understand Visual Basic object-oriented programming, learn to use various .NET Framework Class Library classes and participate in extensive laboratory assignments for hands on experience. The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. After taking this course, students will be prepared to take VB300—*Advanced Visual Basic 2008*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites

- Object-oriented programming in a language such as C#, C++ or Java

Introduction

Review of Visual Basic syntax and concepts presented in the following chapters of *Visual Basic 2008 How to Program*:

- Chapter 2—Introduction to the Visual Basic Express 2008 IDE
- Chapter 3—Introduction to Visual Basic Programming
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Methods: A Deeper Look
- Chapter 8—Arrays

Introduction to LINQ and Generic Collections

- Querying an Array Using LINQ
- Introduction to Collections
- Querying a Generic Collection Using LINQ
- Deitel LINQ Resource Center

Classes and Objects: A Deeper Look

- `Time` Class Case Study
- Class Scope
- Object Initializers
- Default and Parameterless Constructors
- `Time` Class Case Study: Overloaded Constructors
- Partial Classes
- Composition
- Using `Me` to Access the Current Object
- Garbage Collection
- `Shared` Class Members
- `Const` and `ReadOnly` Members
- Object Browser

- `Time` Class Case Study: Creating Class Libraries
- `Time` Class Case Study: Extension Methods
- Delegates
- Lambda Expressions
- Anonymous Types

Object-Oriented Programming: Inheritance
- Base Classes and Derived Classes
- `Protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using Protected Instance Variables
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using Private Instance Variables
- Constructors in Derived Classes
- Software Engineering with Inheritance
- Class `Object`
- `Friend` Members

Object-Oriented Programming: Polymorphism
- Polymorphism Examples
- Demonstrating Polymorphic Behavior
- Abstract Classes and Methods
- Case Study: Payroll System Class Hierarchy Using Polymorphism
- Demonstrating Polymorphic Processing, Expression `TypeOf…Is`, `TryCast` and Downcasting
- Summary of the Allowed Assignments between Base-Class and Derived-Class Variables
- `NotOverridable` Methods and `NotInheritable` Classes
- Case Study: Creating and Using Interfaces
- Common Interfaces of the .NET Framework Class Library

Exception Handling
- Exception-Handling Overview
- Example: Divide by Zero Without Exception Handling
- Example: Handling `DivideByZeroException`s and `FormatException`s
- Enclosing Code in a Try Block
- Catching Exceptions
- Uncaught Exceptions
- Termination Model of Exception Handling
- Flow of Control When Exceptions Occur
- .NET Exception Hierarchy
- Determining Which Exceptions a Method Throws
- `Finally` Block
- Exception Properties
- User-Defined Exception Classes

Graphical User Interfaces with Windows Forms: Part 1
- Windows Forms

- Event Handling
- A Simple Event-Driven GUI
- Another Look at the Visual Studio Generated Code
- Delegates and the Event-Handling Mechanism
- Other Ways to Create Event Handlers
- Locating Event Information
- Control Properties and Layout
- `Labels`, `TextBoxes` and `Buttons`
- `GroupBoxes` and `Panels`
- `CheckBoxes` and `RadioButtons`
- `PictureBoxes`
- `ToolTips`
- `NumericUpDown` Control
- Mouse-Event Handling

GUI with Windows Presentation Foundation
- Windows Presentation Foundation (WPF)
- XML Basics
- Structuring Data
- XML Namespaces
- Declarative GUI Programming Using XAML
- Creating a WPF Application in Visual Basic Express
- Laying Out Controls
- General Layout Principles
- Layout in Action
- Event Handling
- Commands and Common Application Tasks
- WPF GUI Customization
- Using Styles to Change the Appearance of Controls
- Customizing Windows
- Defining a Control's Appearance with Control Templates
- Data-Driven GUIs with Data Binding

Graphics and Multimedia
- Controlling Fonts
- Basic Shapes
- Polygons and Polylines
- Brushes
- Transforms
- WPF Customization: A Television GUI
- Animations

Files and Streams
- Classes `File` and `Directory`
- Creating a Sequential-Access Text File
- Reading Data from a Sequential-Access Text File
- Serialization
- Creating a Sequential-Access File Using Object Serialization
- Reading and Deserializing Data from a Sequential-Access File

Data Structures and Generic Collections

- Generic Collections Overview
- Working with Generic Collections
- Generic Class `LinkedList`
- Generic Class `Dictionary`
- Generic Class `SortedDictionary`
- Introduction to Generic Classes and Methods
- Motivation for Generic Methods
- Generic Method Implementation
- Type Constraints
- Overloading Generic Methods
- Generic Classes
- Primitive-Type Structures; Boxing and Unboxing

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study

An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual Basic 2008 How to Program* (ISBN: 013605305X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

VB300—Advanced Visual Basic 2008

Overview
This 5-day, lecture and laboratory course presents enterprise and web-application development techniques in Visual Basic 2008. Key topics include Language Integrated Query (LINQ), XML, LINQ to XML, Windows Presentation Foundation (WPF), Manipulating databases with LINQ to SQL, SQL Server Express, ASP.NET 3.5, ASP.NET AJAX, building web services with Windows Communication Foundation (WCF), consuming WCF web services and building Rich Internet Applications (RIAs) with Silverlight. The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- VB200—*Visual Basic 2008 for Java, C# or C++ Programmers*, or
- VB112—*Visual Basic 2008 for VB6, C or COBOL Programmers*, or
- VB102—*Introduction to Visual Basic 2008 for Non-Programmers: Part 2*, plus practical Visual Basic 2008 programming experience, or
- Equivalent Visual Basic programming experience

Introduction
- Review of Chapter 9—Introduction to LINQ and Generic Collections
- Review of Chapter 16—GUI with Windows Presentation Foundation

XML and LINQ to XML
- W3C XML Schema Documents
- Extensible Stylesheet Language and XSL Transformations
- LINQ to XML: Document Object Model (DOM)
- LINQ to XML: XML Axis Properties
- LINQ to XML: XML Literals and Embedded Expressions
- XSLT with Class XslCompiledTransform

Database, SQL and LINQ to SQL
- Relational Databases
- Relational Database Overview: Books Database
- Overview Structured Query Language (SQL)
- LINQ to SQL (SQL in this case means Microsoft's SQL Server DBMS)
- Creating LINQ to SQL Classes
- Creating Data Bindings
- More Complex LINQ Queries and Data Binding
- Retrieving Data from Multiple Tables with LINQ
- Creating a Master/Detail View Application
- Programming with LINQ to SQL: Address-Book Case Study

ASP.NET 3.5 and ASP.NET AJAX
- Simple HTTP Transactions
- Multitier Application Architecture
- Creating and Running a Simple Web-Form Example
- Examining an ASPX File
- Examining a Code-Behind File
- Relationship Between an ASPX File and a Code-Behind File

- How the Code in an ASP.NET Web Page Executes
- Examining the XHTML Generated by an ASP.NET Application
- Building an ASP.NET Web Application
- Web Controls
    - Text and Graphics Controls
    - AdRotator Control
    - Validation Controls
- Session Tracking
    - Cookies
    - Session Tracking with `HttpSessionState`
- Case Study: Guestbook Application—Connecting to a Database in ASP.NET
- Case Study: Secure Books Database Application
- ASP.NET AJAX
    - Traditional Web Applications
    - Ajax Web Applications
    - Examining an ASP.NET AJAX Application
- New ASP.NET 3.5 Data Controls

Windows Communication Foundation (WCF) Web Services

- WCF Services Basics
- Simple Object Access Protocol (SOAP)
- Representational State Transfer (REST)
- JavaScript Object Notation (JSON)
- Publishing and Consuming SOAP-Based WCF Web Services
    - Creating a WCF Web Service
    - Code for the `WelcomeSOAPXMLService`
    - Building a SOAP WCF Web Service
    - Deploying the `WelcomeSOAPXMLService`
    - Creating a Client to Consume the `WelcomeSOAPXMLService`
    - Consuming the `WelcomeSOAPXMLService`
- Publishing and Consuming REST-Based XML Web Services
    - Creating a REST-Based XML WCF Web Service
    - Consuming a REST-Based XML WCF Web Service
- Publishing and Consuming REST-Based JSON Web Services
    - Creating a REST-Based JSON WCF Web Service
    - Consuming a REST-Based JSON WCF Web Service
- Blackjack Web Service
    - Using Session Tracking in a SOAP-Based WCF Web Service
    - Creating a Blackjack Web Service
    - Consuming the Blackjack Web Service
- Airline Reservation Web Service
    - Database Access and Invoking a Service from ASP.NET
- Equation Generator: Returning User-Defined Types
    - Creating the REST-Based XML EquationGenerator Web Service
    - Consuming the REST-Based XML EquationGenerator Web Service
    - Creating the REST-Based JSON WCF EquationGenerator Web Service
    - Consuming the REST-Based JSON WCF EquationGenerator Web Service

Silverlight, Rich Internet Applications and Multimedia

- Platform Overview
- Silverlight Demos

- Silverlight Runtime and Tools Installation
- Building a Silverlight `WeatherViewer` Application
- GUI Layout
- Obtaining and Displaying Weather Forecast Data
- Custom Controls
- Animations and the `FlickrViewer`
- Images and Deep Zoom
    - Getting Started With Deep Zoom Composer
    - Creating a Silverlight Deep Zoom Application
- Audio and Video
- Isolated Storage

Price

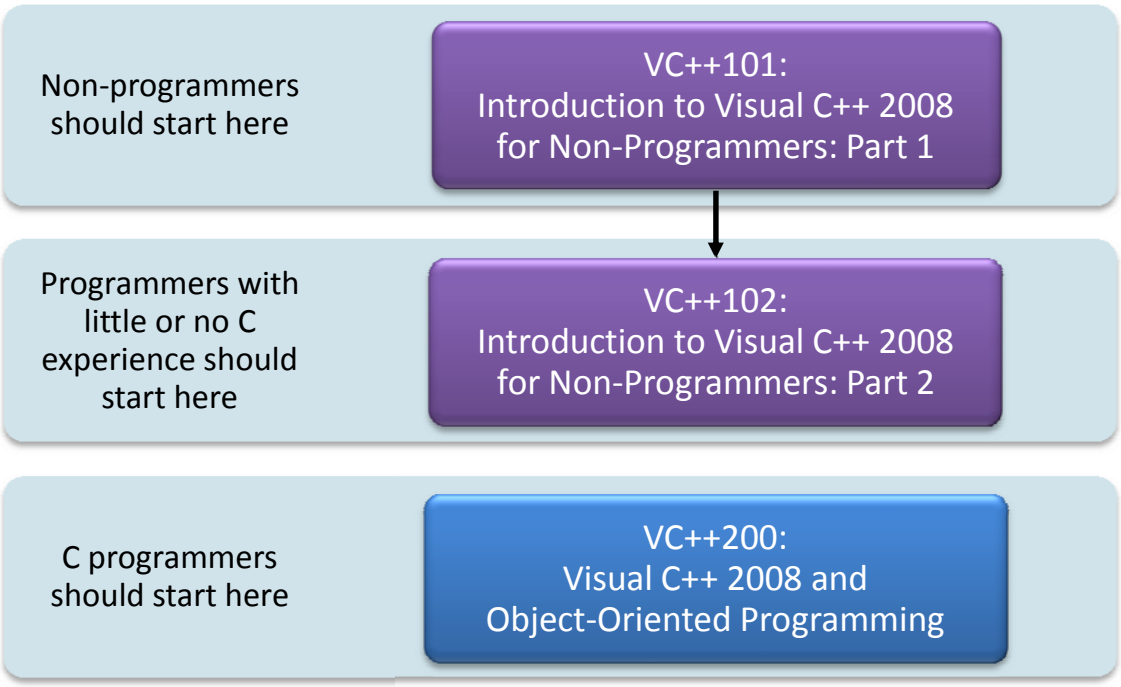
- $13,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual Basic 2008 How to Program* (ISBN: 013605305X)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Visual C++ 2008 Programming Curriculum

Overview

Visual C++ is a powerful computer programming language that is appropriate for technically oriented people with little or no programming experience and for experienced programmers to use in building substantial information systems. Visual C++ is popular for high-performance applications and for building high-performance software components used by other .NET applications.

Deitel & Associates, Inc. is the world's leading programming language textbook/professional book authors and our courses are based on our best-selling books.



VC++101—Introduction to Visual C++ 2008 for Non-Programmers: Part 1

Overview
This 5-day, lecture-and-lab course teaches fundamental C++ concepts such as algorithmic thinking, problem solving, introduction to classes and objects, control structures (`if`, `if…else`, `switch`, `while`, `do…while`, `for`), data types, operators, input/output, functions (user-defined and library), single-subscripted arrays and strings. After taking this course, students will be prepared to take VC++102—*Introduction to Visual C++ 2008 for Non-Programmers: Part 2*. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Our introductory classes emphasize achieving program clarity through the proven software-development techniques. Nonprogrammers learn skills that underlie good programming through dozens of complete, working programs—we call this the live-code approach. This hands-on approach rapidly builds the confidence of new programmers, because all programming concepts are presented in the context in which they will be used.

Prerequisites
- General familiarity with your operating system environment
- Ability to create and edit text files

Dive Into® Visual C++® 2008 Express
- Overview of the Visual Studio 2008 IDE
- Menu Bar and Toolbars
- Navigating the Visual Studio 2008 IDE
- Solution Explorer
- Properties Window
- Using Help

Introduction to Visual C++ Programming
- First Program in Visual C++: Printing a Line of Text
- Modifying Our First Visual C++ Program
- Another Visual C++ Program: Adding Integers
- Memory Concepts
- Arithmetic
- Decision Making: Equality and Relational Operators

Introduction to Classes and Objects
- Classes, Objects, Member Functions and Data Members
- Overview of the Chapter Examples
- Defining a Class with a Member Function
- Defining a Member Function with a Parameter
- Data Members, *set* Functions and *get* Functions
- Initializing Objects with Constructors
- Placing a Class in a Separate File for Reusability
- Separating Interface from Implementation
- Validating Data with *set* Functions
- Defining a Managed Class with Member Functions in C++/CLI
- Instance Variables and Properties in C++/CLI

Control Statements: Part 1
- Algorithms
- Pseudocode
- Control Structures

- `if` Selection Statement
- `if…else` Double-Selection Statement
- `while` Repetition Statement
- Formulating Algorithms: Counter-Controlled Repetition
- Formulating Algorithms: Sentinel-Controlled Repetition
- Formulating Algorithms: Nested Control Statements
- Assignment Operators
- Increment and Decrement Operators

Control Statements: Part 2
- Essentials of Counter-Controlled Repetition
- `for` Repetition Statement
- Examples Using the `for` Statement
- `do…while` Repetition Statement
- `switch` Multiple-Selection Statement
- `break` and `continue` Statements
- Logical Operators
- Confusing the Equality (==) and Assignment (=) Operators

Functions
- Program Components in Visual C++
- `Math` Library Functions
- Function Definitions with Multiple Parameters
- Function Prototypes and Argument Coercion
- C++ Standard Library Header Files
- Case Study: Random Number Generation
- Case Study: Game of Chance; Introducing enum
- Storage Classes
- Scope Rules
- Function-Call Stack and Activation Records
- Functions with Empty Parameter Lists

Arrays and Vectors
- Arrays
- Declaring Arrays
- Examples Using Arrays
- Declaring an Array and Using a Loop to Initialize the Array's Elements
- Initializing an Array in a Declaration with an Initializer List
- Specifying an Array's Size with a Constant Variable and Setting Array Elements with Calculations
- Summing the Elements of an Array
- Using Bar Charts to Display Array Data Graphically
- Using the Elements of an Array as Counters
- Using Arrays to Summarize Survey Results
- Static Local Arrays and Automatic Local Arrays
- Passing Arrays to Functions
- Case Study: Class `GradeBook` Using an Array to Store Grades

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C++ 2008 How to Program, 2/e* (ISBN: 0136151574)

- To order the books at the GSA rates, please contact:
  Kathryn Bass, Government Sales Executive
  Pearson Education
  Phone: 703-404-9194
  Fax 703-404-9195
  Kathryn.Bass@Pearson.com
  GSA Purchases: www.pearsongovernmentsales.com
  Pearson GSA Contract # GS14F8023A

VC++102—Introduction to Visual C++ 2008 for Non-Programmers: Part 2

Overview
This 5-day, lecture-and-lab course covers in-line functions, overloaded functions, function templates, default arguments, references, const qualifier, arrays, multiple-subscripted arrays, pointers, pointer arithmetic, function pointers, strings, optimizing memory use and improving execution performance. The course can be customized to the Ordering Activity's needs at no additional charge.

Prerequisites
- VC++101—*Introduction to Visual C++ 2008 for Non-Programmers: Part 1* or equivalent programming experience

Introduction
Review of C++ syntax and concepts presented in the following chapters of *Visual C++ 2008 How to Program, 2/e* as part of the course *Introduction to Visual C++ 2008 for Non-Programmers: Part 1*.
- Chapter 2—Dive Into® Visual C++® 2008 Express
- Chapter 3—Introduction to Visual C++ Programming
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Functions and an Introduction to Recursion
- Chapter 8—Arrays and Vectors

Functions
- Inline Functions
- References and Reference Parameters
- Default Arguments
- Unary Scope Resolution Operator
- Function Overloading
- Function Templates
- Enumerations in C++/CLI

Arrays and Vectors
- Multidimensional Arrays
- Case Study: Class `GradeBook` Using a Two-Dimensional Array
- Introduction to C++ Standard Library Class Template `vector`
- Introduction to Managed Arrays with C++/CLI
- `for each` Statement
- Multidimensional Arrays in C++/CLI

Pointers and Pointer-Based Strings
- Pointer Variable Declarations and Initialization
- Pointer Operators
- Passing Arguments to Functions by Reference with Pointers
- Using `const` with Pointers
- `sizeof` Operator
- Pointer Expressions and Pointer Arithmetic
- Relationship Between Pointers and Arrays
- Arrays of Pointers
- Case Study: Card Shuffling and Dealing Simulation
- Function Pointers
- Introduction to Pointer-Based String Processing

- Fundamentals of Characters and Pointer-Based Strings
- String-Manipulation Functions of the String-Handling Library
- Introduction to C++/CLI Handles
- Passing Arguments to Functions by Reference with Handles
- Tracking References and References to Handles
- Interior Pointers

Classes: A Deeper Look, Part 1
- `Time` Class Case Study
- Class Scope and Accessing Class Members
- Separating Interface from Implementation
- Access Functions and Utility Functions
- `Time` Class Case Study: Constructors with Default Arguments
- Destructors
- When Constructors and Destructors Are Called
- `Time` Class Case Study: A Subtle Trap—Returning a Reference to a private Data Member
- Default Memberwise Assignment
- Class View and Object Browser

Classes: A Deeper Look, Part 2
- `const` (Constant) Objects and const Member Functions
- Composition: Objects as Members of Classes
- `friend` Functions and friend Classes
- Using the `this` Pointer
- Dynamic Memory Management with Operators new and delete
- `static` Class Members
- Container Classes and Iterators
- Proxy Classes
- `const` and `friend` in C++/CLI
- Dynamic Memory Management in C++/CLI
- Stack Semantics in C++/CLI
- Finalizers
- Value Types vs. Reference Types in C++/CLI
- Boxing and Unboxing in C++/CLI
- Indexers

Operator Overloading; String and Array Objects
- Fundamentals of Operator Overloading
- Restrictions on Operator Overloading
- Operator Functions as Class Members vs. Global Functions
- Overloading Stream Insertion and Stream Extraction Operators
- Overloading Unary Operators
- Overloading Binary Operators
- Case Study: `Array` Class

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C++ 2008 How to Program, 2/e* (ISBN: 0136151574)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education

Phone: 703-404-9194
Fax 703-404-9195
Kathryn.Bass@Pearson.com
GSA Purchases: www.pearsongovernmentsales.com
Pearson GSA Contract # GS14F8023A

VC++200—Visual C++ 2008 and Object Oriented Programming

Overview
An intensive, 5-day, lecture-and-lab Visual C++ 2008 and object-oriented programming course for practicing C (or other high-level language) programmers. Visual C++ 2008 offers all the features of standard C++ and the ability to work with the Visual C++-specific managed code capabilities of the C++/CLI (Common Language Infrastructure). This course covers Visual C++ 2008's features for data abstraction and encapsulation that enable programmers to create new types, and capabilities for object-oriented programming with inheritance and polymorphism. The course clearly explains the Visual C++ 2008 programming language and contains detailed walkthroughs of many Visual C++ programs. Students become proficient in Visual C++, understand the object-oriented paradigm, and participate in challenging hands-on laboratory assignments. Solutions are provided for the laboratory exercises. 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- Programming in C or another high-level language, or

Introduction
Review of C++ syntax and concepts presented in the following chapters of *Visual C++ 2008 How to Program, 2/e*.
- Chapter 2—Dive Into® Visual C++® 2008 Express
- Chapter 3—Introduction to Visual C++ Programming
- Chapter 4—Introduction to Classes and Objects
- Chapter 5—Control Statements: Part 1
- Chapter 6—Control Statements: Part 2
- Chapter 7—Functions and an Introduction to Recursion
- Chapter 8—Arrays and Vectors
- Chapter 9—Pointers and Pointer-Based Strings

Classes: A Deeper Look, Part 1
- `Time` Class Case Study
- Class Scope and Accessing Class Members
- Separating Interface from Implementation
- Access Functions and Utility Functions
- `Time` Class Case Study: Constructors with Default Arguments
- Destructors
- When Constructors and Destructors Are Called
- `Time` Class Case Study: A Subtle Trap—Returning a Reference to a private Data Member
- Default Memberwise Assignment
- Class View and Object Browser

Classes: A Deeper Look, Part 2
- const (Constant) Objects and const Member Functions
- Composition: Objects as Members of Classes
- `friend` Functions and friend Classes
- Using the `this` Pointer
- Dynamic Memory Management with Operators new and delete
- `static` Class Members
- Proxy Classes

- `const` and `friend` in C++/CLI
- Dynamic Memory Management in C++/CLI
- Stack Semantics in C++/CLI
- Finalizers
- Value Types vs. Reference Types in C++/CLI
- Boxing and Unboxing in C++/CLI
- Indexers

Operator Overloading; String and Array Objects
- Fundamentals of Operator Overloading
- Restrictions on Operator Overloading
- Operator Functions as Class Members vs. Global Functions
- Overloading Stream Insertion and Stream Extraction Operators
- Overloading Unary Operators
- Overloading Binary Operators
- Case Study: `Array` Class
- Converting between Types
- Case Study: `String` Class
- Overloading ++ and `--`
- Case Study: A `Date` Class
- Standard Library Class string
- `explicit` Constructors
- C++/CLI Operators and Constructors

Object-Oriented Programming: Inheritance
- Base Classes and Derived Classes
- `protected` Members
- Relationship between Base Classes and Derived Classes
- Creating and Using a `CommissionEmployee` Class
- Creating a `BasePlusCommissionEmployee` Class Without Using Inheritance
- Creating a `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using `protected` Data
- `CommissionEmployee–BasePlusCommissionEmployee` Inheritance Hierarchy Using `private` Data
- Constructors and Destructors in Derived Classes
- `public`, `protected` and `private` Inheritance
- Inheritance in C++/CLI

Object-Oriented Programming: Polymorphism
- Polymorphism Examples
- Relationships Among Objects in an Inheritance Hierarchy
- Invoking Base-Class Functions from Derived-Class Objects
- Aiming Derived-Class Pointers at Base-Class Objects
- Derived-Class Member-Function Calls via Base-Class Pointers
- Virtual Functions
- Summary of the Allowed Assignments Between Base-Class and Derived-Class Objects and Pointers
- Type Fields and `switch` Statements
- Abstract Classes and Pure `virtual` Functions
- Case Study: Payroll System Using Polymorphism

- Polymorphism, Virtual Functions and Dynamic Binding “Under the Hood”
- Case Study: Payroll System Using Polymorphism and Runtime Type Information with Downcasting, `dynamic_cast`, `typeid` and `type_info`
- Virtual Destructors
- Polymorphism in C++/CLI

Templates and Generics
- Function Templates
- Overloading Function Templates
- Class Templates
- Nontype Parameters and Default Types for Class Templates
- Templates in C++/CLI
- .NET Generics in C++/CLI
- Generic Type Constraints
- Contrasting Templates and Generics

Exception Handling
- Exception-Handling Overview
- Example: Divide by Zero Without Exception Handling
- Example: Handling an Attempt to Divide by Zero
- When to Use Exception Handling
- Rethrowing an Exception
- Processing Unexpected Exceptions
- Stack Unwinding
- Constructors, Destructors and Exception Handling
- Exceptions and Inheritance
- Processing new Failures
- Class `auto_ptr` and Dynamic Memory Allocation
- Standard Library Exception Hierarchy
- Other Error-Handling Techniques
- .NET Exception Hierarchy with C++/CLI
- Classes ApplicationException and SystemException
- Determining Which Exceptions a Function Throws
- `finally` Block in C++/CLI
- Exception Properties in C++/CLI
- User-Defined Exception Classes in .NET

Files and Streams in .NET
- Files and Streams
- Classes File and Directory
- Creating a Sequential-Access Text File
- Reading Data from a Sequential-Access Text File
- Serialization
- Creating a Sequential-Access File Using Object Serialization
- Reading and Deserializing Data from a Sequential-Access Text File

Standard Template Library (STL)
- Introduction to the Standard Template Library (STL)
- Introduction to Containers
- Introduction to Iterators

- Introduction to Algorithms
- Sequence Containers
- Associative Containers
- Container Adapters
- Algorithms
- Function Objects
- Introduction to STL/CLR

Collections
- Collections Overview
- Class Array and Enumerators
- Nongeneric Collections
- Generic Collections

Optional Object-Oriented Design (OOD) with the Unified Modeling Language (UML) Case Study
An optional topic for our object-oriented programming classes is our case study on object-oriented design using the UML in which we design and fully implement the software for a simple automated teller machine (ATM). We introduce a subset of the UML 2.0, then guide the reader through an end-to-end first object-oriented design and implementation experience which ends with a walkthrough of the complete code.

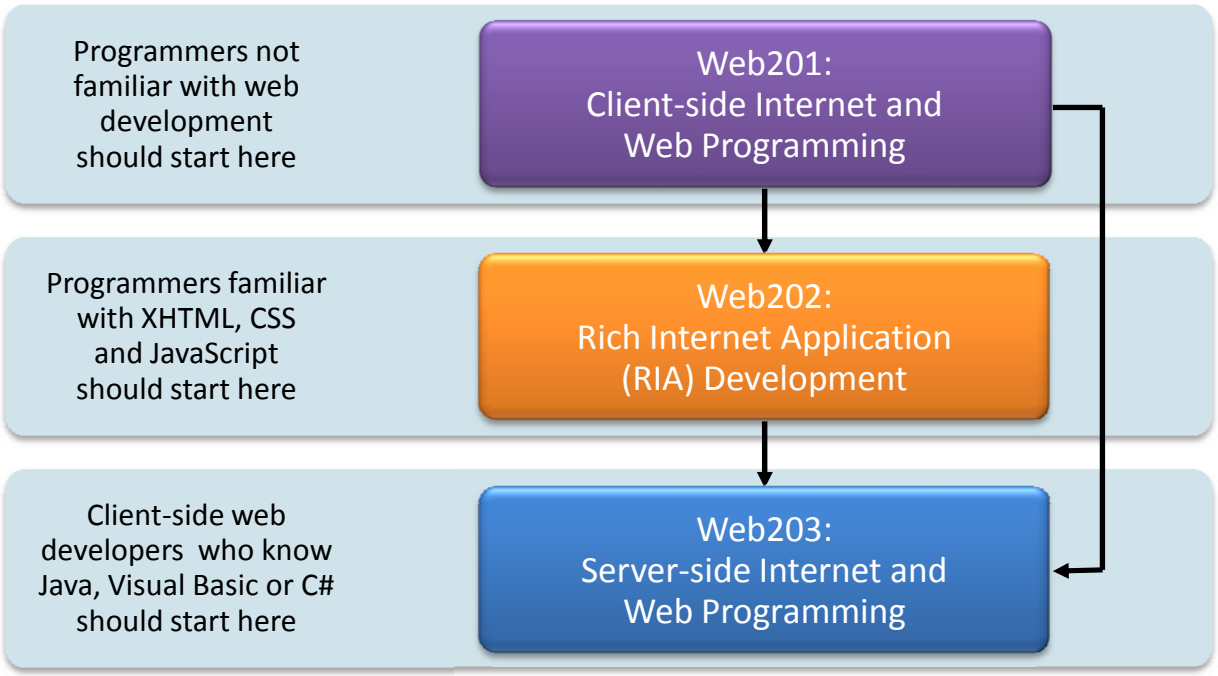
Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Visual C++ 2008 How to Program, 2/e* (ISBN: 0136151574)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Internet & Web Programming Curriculum

Overview

In these classes, we present a number of powerful software technologies that will enable you to build systems that can integrate Internet and web components, and remote databases. We present the "client-side" and "server-side" of web programming. For the client side we present a carefully paced introduction to using the popular JavaScript language and the closely related technologies of XHTML (Extensible HyperText Markup Language), CSS (Cascading Style Sheets) and the DOM (Document Object Model). Novices will find that the material in the JavaScript chapters presents a solid foundation for the deeper treatment of scripting in the second class—*Rich Internet Application (RIA) Development*—which covers XML, AJAX, Adobe Flash, Adobe Flex and Microsoft Silverlight. The third class concentrates on using technologies such as web servers, databases (integrated collections of data), PHP, Ruby on Rails, ASP.NET, ASP.NET Ajax and JavaServer Faces (JSF) to build the server side of web-based applications. These portions of applications typically run on "heavy-duty" computer systems on which organizations' business-critical websites reside. By mastering the technologies in these courses, you'll be able to build substantial web-based, client/server, database-intensive, "multitier" applications.

Deitel & Associates, Inc. is the world's leading programming language textbook/professional book authors and our courses are based on our best-selling books.



Web201—Client-side Internet and Web Programming

Overview
This 5-day, lecture and laboratory course focuses on the client-side of web-application development. Key topics include eXtensible HyperText Markup Language (XHTML), Cascading Style Sheets (CSS), JavaScript (variables, control statements, functions, arrays, objects), the Document Object Model (DOM), event handling, eXtensible Markup Language (XML), Really Simple Syndication (RSS) and Asynchronous JavaScript and XML (AJAX). The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- Programming in a high-level programming language
- General familiarity with the web and web browsing

Introduction
- Overview of *Internet and World Wide Web How to Program, 4/e*

Web Browser Basics: Internet Explorer and Firefox
- This chapter's contents are discussed throughout the course

Introduction to XHTML
- Editing XHTML
- First XHTML Example
- W3C XHTML Validation Service
- Headings
- Linking
- Images
- Special Characters and Horizontal Rules
- Lists
- Tables
- Forms
- Internal Linking
- meta Elements

Cascading Style Sheets™ (CSS)
- Inline Styles
- Embedded Style Sheets
- Conflicting Styles
- Linking External Style Sheets
- Positioning Elements
- Backgrounds
- Element Dimensions
- Box Model and Text Flow
- Media Types
- Building a CSS Drop-Down Menu
- User Style Sheets
- CSS 3

JavaScript: Introduction to Scripting
- Simple Program: Displaying a Line of Text in a Web Page
- Modifying Our First Program

- Obtaining User Input with prompt Dialogs
- Dynamic Welcome Page
- Decision Making: Equality and Relational Operators

JavaScript: Control Statements I
- `if` Selection Statement
- `if…else` Selection Statement
- `while` Repetition Statement
- Assignment Operators
- Increment and Decrement Operators

JavaScript: Control Statements II
- `for` Repetition Statement
- `switch` Multiple-Selection Statement
- `do…while` Repetition Statement
- Logical Operators

JavaScript: Functions
- Programmer-Defined Functions
- Function Definitions
- Random Number Generation
- Example: Game of Chance
- Another Example: Random Image Generator
- Scope Rules
- JavaScript Global Functions

JavaScript: Arrays
- Arrays
- Declaring and Allocating Arrays
- Examples Using Arrays
- Random Image Generator Using Arrays
- References and Reference Parameters
- Passing Arrays to Functions
- Sorting Arrays
- Searching Arrays
- Multidimensional Arrays
- Building an Online Quiz

JavaScript: Objects
- Introduction to Object Technology
- `Math` Object
- `String` Object
- Fundamentals of Characters and Strings
- Methods of the `String` Object
- Character-Processing Methods
- Searching Methods
- Splitting Strings and Obtaining Substrings
- XHTML Markup Methods
- `Date` Object
- `Boolean` and `Number` Objects
- `document` Object
- `window` Object

- Using Cookies
- Final JavaScript Example
- Using JSON to Represent Objects

Document Object Model (DOM): Objects and Collections
- Modeling a Document: DOM Nodes and Trees
- Traversing and Modifying a DOM Tree
- DOM Collections
- Dynamic Styles
- Summary of the DOM Objects and Collections

JavaScript: Events
- Registering Event Handlers
- Event `onload`
- Event `onmousemove`, the event `Object` and `this`
- Rollovers with `onmouseover` and `onmouseout`
- Form Processing with `onfocus` and `onblur`
- More Form Processing with `onsubmit` and `onreset`
- Event Bubbling

XML and RSS
- XML Basics
- Structuring Data
- XML Namespaces
- Document Type Definitions (DTDs)
- W3C XML Schema Documents
- XML Vocabularies
- Extensible Stylesheet Language and XSL Transformations
- Document Object Model (DOM)
- RSS

Ajax-Enabled Rich Internet Applications
- Traditional Web Applications vs. Ajax Applications
- Rich Internet Applications (RIAs) with Ajax
- History of Ajax
- "Raw" Ajax Example Using the `XMLHttpRequest` Object
- Using XML and the DOM
- Creating a Full-Scale Ajax-Enabled Application
- Dojo Toolkit

Price
- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Internet and World Wide Web How to Program, 4/e* (ISBN: 0131752421)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194 Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com (Pearson GSA Contract # GS14F8023A)

Web202—Rich Internet Application (RIA) Development

Overview
This 5-day, lecture-and-lab course focuses on technologies for building Rich Internet Applications (RIAs) that run in web browsers. Topics include eXtensible Markup Language (XML), Really Simple Syndication (RSS), Asynchronous JavaScript and XML (AJAX), Adobe Flash, Adobe Flex and Microsoft Silverlight. Solutions are provided for lab exercises. Approximately 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- Programming at the level of Web201—*Client-side Internet and Web Programming*
- Programming in JavaScript for the XML, AJAX, Flash and Flex material; programming in Visual Basic (.NET version) or Visual C# for the Silverlight Material

XML and RSS
- XML Basics
- Structuring Data
- XML Namespaces
- Extensible Stylesheet Language and XSL Transformations
- Document Object Model (DOM)
- RSS

Ajax-Enabled Rich Internet Applications
- Traditional Web Applications vs. Ajax Applications
- Rich Internet Applications (RIAs) with Ajax
- History of Ajax
- "Raw" Ajax Example Using the XMLHttpRequest Object
- Using XML and the DOM
- Creating a Full-Scale Ajax-Enabled Application
- Dojo Toolkit

Adobe® Flash® CS3
- Flash Movie Development
- Learning Flash with Hands-On Examples
- Creating a Shape with the Oval Tool
- Adding Text to a Button
- Converting a Shape into a Symbol
- Editing Button Symbols
- Adding Keyframes
- Adding Sound to a Button
- Verifying Changes with Test Movie
- Adding Layers to a Movie
- Animating Text with Tweening
- Adding a Text Field
- Adding ActionScript
- Publishing Your Flash Movie
- Creating Special Effects with Flash
- Importing and Manipulating Bitmaps
- Creating an Advertisement Banner with Masking
- Adding Online Help to Forms
- Creating a Website Splash Screen
- ActionScript

Adobe® Flash® CS3: Building an Interactive Game

- Object-Oriented Programming
- Objects in Flash
- Cannon Game: Preliminary Instructions and Notes
- Adding a Start Button
- Creating Moving Objects
- Adding the Rotating Cannon
- Adding the Cannonball
- Adding Sound and Text Objects to the Movie
- Adding the Time Counter
- Detecting a Miss
- Adding Collision Detection
- Finishing the Game

Adobe® Flex™ 2 and Rich Internet Applications

- Flex Platform Overview
- Creating a Simple User Interface
- Accessing XML Data from Your Application
- Interacting with Server-Side Applications
- Customizing Your User Interface
- Creating Charts and Graphs
- Connection-Independent RIAs on the Desktop:
- Adobe Integrated Runtime (AIR)

Microsoft® Silverlight™ and Rich Internet Applications

- Platform Overview
- Silverlight Demos
- Silverlight Runtime and Tools Installation
- Building a Silverlight `WeatherViewer` Application
- GUI Layout
- Obtaining and Displaying Weather Forecast Data
- Custom Controls
- Animations and the `FlickrViewer`
- Images and Deep Zoom
    - Getting Started With Deep Zoom Composer
    - Creating a Silverlight Deep Zoom Application
- Audio and Video
- Isolated Storage

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Internet and World Wide Web How to Program, 4/e* (ISBN: 0131752421)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

Web203—Server-side Internet and Web Programming

Overview
This 5-day, lecture and laboratory course surveys various technologies for building the server side of web applications. Key topics include web servers, database access, PHP, Ruby/Ruby on Rails, ASP.NET and ASP.NET AJAX, JavaServer Faces and web services. The course offers extensive hands-on laboratory experience. Solutions are provided for laboratory exercises. The course includes approximately 60% lecture and 40% laboratory exercises. The course can be customized to the Ordering Activity's specific needs at no additional charge.

Prerequisites
- Web201—*Client-side Internet and Web Programming*, or equivalent programming experience, or
- Object-oriented programming experience in current programming languages, such as Visual Basic, Visual C# or Java.

Web Servers (IIS and Apache)
- HTTP Transactions
- Multitier Application Architecture
- Client-Side Scripting versus Server-Side Scripting
- Accessing Web Servers
- Microsoft Internet Information Services (IIS)
- Microsoft Internet Information Services (IIS) 5.1 and 6.0
- Microsoft Internet Information Services (IIS) 7.0
- Apache HTTP Server
- Requesting Documents

Database: SQL, MySQL, ADO.NET 2.0 and Java DB
- Relational Databases
- Relational Database Overview: A books Database
- SQL
- Basic SELECT Query
- WHERE Clause
- ORDER BY Clause
- Combining Data from Multiple Tables: INNER JOIN
- INSERT Statement
- UPDATE Statement
- DELETE Statement
- MySQL
- Instructions for Installing MySQL
- Instructions for Setting Up a MySQL User Account
- Creating a Database in MySQL
- ADO.NET Object Model
- Java DB/Apache Derby

PHP
- PHP Basics
- String Processing and Regular Expressions
- Comparing Strings
- Regular Expressions
- Form Processing and Business Logic

- Connecting to a Database
- Using Cookies
- Dynamic Content
- Operator Precedence Chart

Ruby on Rails
- Ruby
- Rails Framework
- ActionController and ActionView
- A Database-Driven Web Application
- Case Study: Message Forum
- Logging In and Logging Out
- Embellishing the Models
- Generating Scaffold Code
- Forum Controller and Forum Views
- Message Controller and Message Views
- Ajax-Enabled Rails Applications
- Script.aculo.us

ASP.NET 2.0 and ASP.NET Ajax
- Creating and Running a Simple Web Form Example
- Examining an ASPX File
- Examining a Code-Behind File
- Relationship Between an ASPX File and a Code-Behind File
- How the Code in an ASP.NET Web Page Executes
- Examining the XHTML Generated by an ASP.NET Application
- Building an ASP.NET Web Application
- Web Controls
- Text and Graphics Controls
- AdRotator Control
- Validation Controls
- Session Tracking
- Cookies
- Session Tracking with HttpSessionState
- Case Study: Connecting to a Database in ASP.NET
- Building a Web Form That Displays Data from a Database
- Modifying the Code-Behind File for the Guestbook Application
- Case Study: Secure Books Database Application
- Examining the Completed Secure Books Database Application
- Creating the Secure Books Database Application
- ASP.NET Ajax

JavaServer™ Faces Web Applications
- Java Web Technologies
- Servlets
- JavaServer Pages
- JavaServer Faces
- Web Technologies in Netbeans
- Creating and Running a Simple Application in Netbeans
- Examining a JSP File

- Examining a Page Bean File
- Event-Processing Life Cycle
- Relationship Between the JSP and Page Bean Files
- Examining the XHTML Generated by a Java Web Application
- Building a Web Application in Netbeans
- JSF Components
- Text and Graphics Components
- Validation Using Validator Components and Custom Validators
- Session Tracking
- Cookies
- Session Tracking with the SessionBean Object

Ajax-Enabled JavaServer™ Faces Web Applications

- Accessing Databases in Web Applications
- Building a Web Application That Displays Data from a Database
- Modifying the Page Bean File for the AddressBook Application
- Ajax-Enabled JSF Components
- AutoComplete Text Field and Virtual Forms
- Configuring Virtual Forms
- JSP File with Virtual Forms and an AutoComplete Text Field
- Providing Suggestions for an AutoComplete Text Field
- Google Maps Map Viewer Component
- Obtaining a Google Maps API Key
- Adding a Map Viewer Component to a Page
- JSP File with a Map Viewer Component
- Page Bean That Displays a Map in the Map Viewer Component

Web Services

- Java Web Services Basics
- Creating, Publishing, Testing and Describing a Web Service
- Creating a Web Application Project and Adding a Web
- Defining the HugeInteger Web Service in Netbeans
- Publishing the HugeInteger Web Service from Netbeans
- Testing the HugeInteger Web Service with Sun Java System
- Application Server's Tester Web page
- Describing a Web Service with the Web Service
- Description Language (WSDL)
- Consuming a Web Service
- Creating a Client in Netbeans to Consume the HugeInteger
- Web Service
- Consuming the HugeInteger Web Service
- SOAP
- Session Tracking in Web Services
- Creating a Blackjack Web Service
- Consuming the Blackjack Web Service
- Consuming a Database-Driven Web Service from a Web Application
- Configuring Java DB in Netbeans and Creating the
- Reservation Database
- Creating a Web Application to Interact with the
- Reservation Web Service

- Passing an Object of a User-Defined Type to a Web Service
- REST-Based Web Services in ASP.NET
- REST-Based Web Service Functionality
- Creating an ASP.NET REST-Based Web Service
- Adding Data Components to a Web Service

Price

- $12,995 lecture fee for up to 20 students maximum.
- Ordering Activity purchases the books, at its own expense, directly from Pearson (the publisher).
    - Materials: *Internet and World Wide Web How to Program, 4/e* (ISBN: 0131752421)
    - To order the books at the GSA rates, please contact:
      Kathryn Bass, Government Sales Executive
      Pearson Education
      Phone: 703-404-9194
      Fax 703-404-9195
      Kathryn.Bass@Pearson.com
      GSA Purchases: www.pearsongovernmentsales.com
      Pearson GSA Contract # GS14F8023A

SIN 132-50 Training Course Price List Summary

Course Number	Course Title	Course Length	Max. # of students	Price (Includes IFF)
Java Programming				
Java101	Introduction to Java for Non-Programmers: Part 1	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Java102	Introduction to Java for Non-Programmers: Part 2	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Java112	Java for Visual Basic, C or COBOL Programmers	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Java200	Java for C++ or C# Programmers	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Java300	Advanced Java	Five (5) Days	20	$13,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
C++ Programming				
C++101	Introduction to C++ for Non-Programmers: Part 1	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
C++102	Introduction To C++ for Non- Programmers: Part 2	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
C++200	C++ and Object Oriented Programming	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
C Programming				
C101	Introduction to C for Non-Programmers: Part 1	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
C102	Introduction to C for Non-Programmers: Part 2	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
C200	C for Programmers	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Visual C# 2008 Programming				
VC#101	Introduction to Visual C# 2008 for Non-Programmers: Part 1	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VC#102	Introduction to Visual C# 2008 for Non-Programmers: Part 2	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VC#112	Visual C# 2008 for Visual Basic, C or COBOL Programmers	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.

VC#200	Visual C# 2008 for Java or C++ Programmers	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VC#300	Advanced Visual C# 2008	Five (5) Days	20	$13,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Visual Basic 2008 Programming				
VB101	Introduction to Visual Basic 2008 for Non-Programmers: Part 1	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VB102	Introduction to Visual Basic 2008 for Non-Programmers: Part 2	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VB112	Visual Basic 2008 for VB6, C or COBOL Programmers	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VB200	Visual Basic 2008 for Java, C# or C++ Programmers	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VB300	Advanced Visual Basic 2008	Five (5) Days	20	$13,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Visual C++ 2008 Programming				
VC++101	Introduction to Visual C++ 2008 for Non-Programmers: Part 1	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VC++102	Introduction To Visual C++ 2008 for Non- Programmers: Part 2	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
VC++200	Visual C++ 2008 and Object Oriented Programming	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Internet and Web Programming				
Web201	Client-side Internet and Web Programming	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Web202	Rich Internet Application (RIA) Development	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.
Web203	Server-side Internet and Web Programming	Five (5) Days	20	$12,995 Lecture Fee. Ordering Activity purchases books directly from Pearson at GSA rates.

USA COMMITMENT TO PROMOTE SMALL BUSINESS PARTICIPATION PROCUREMENT PROGRAMS

PREAMBLE

Deitel & Associates, Inc. provides commercial products and services to ordering activities. We are committed to promoting participation of small, small disadvantaged and women-owned small businesses in our contracts. We pledge to provide opportunities to the small business community through reselling opportunities, mentor-protégé programs, joint ventures, teaming arrangements, and subcontracting.

COMMITMENT

To actively seek and partner with small businesses.

To identify, qualify, mentor and develop small, small disadvantaged and women-owned small businesses by purchasing from these businesses whenever practical.

To develop and promote company policy initiatives that demonstrate our support for awarding contracts and subcontracts to small business concerns.

To undertake significant efforts to determine the potential of small, small disadvantaged and women-owned small business to supply products and services to our company.

To insure procurement opportunities are designed to permit the maximum possible participation of small, small disadvantaged, and women-owned small businesses.

To attend business opportunity workshops, minority business enterprise seminars, trade fairs, procurement conferences, etc., to identify and increase small businesses with whom to partner.

To publicize in our marketing publications our interest in meeting small businesses that may be interested in subcontracting opportunities.

We signify our commitment to work in partnership with small, small disadvantaged and women-owned small businesses to promote and increase their participation in ordering activity contracts. To accelerate potential opportunities please contact

Abbey Deitel
Phone: 978 823 0130
abbey.deitel@deitel.com
Fax: 978 823 0132

BEST VALUE
BLANKET PURCHASE AGREEMENT
FEDERAL SUPPLY SCHEDULE

(Insert Customer Name)

In the spirit of the Federal Acquisition Streamlining Act (ordering activity) and (Contractor) enter into a cooperative agreement to further reduce the administrative costs of acquiring commercial items from the General Services Administration (GSA) Federal Supply Schedule Contract(s) _____.

Federal Supply Schedule contract BPAs eliminate contracting and open market costs such as: search for sources; the development of technical documents, solicitations and the evaluation of offers. Teaming Arrangements are permitted with Federal Supply Schedule Contractors in accordance with Federal Acquisition Regulation (FAR) 9.6.

This BPA will further decrease costs, reduce paperwork, and save time by eliminating the need for repetitive, individual purchases from the schedule contract. The end result is to create a purchasing mechanism for the ordering activity that works better and costs less.

Signatures

Ordering Activity	Date	Contractor	Date

BPA NUMBER_____

(CUSTOMER NAME)

BLANKET PURCHASE AGREEMENT

Pursuant to GSA Federal Supply Schedule Contract Number(s)_____, Blanket Purchase Agreements, the Contractor agrees to the following terms of a Blanket Purchase Agreement (BPA) EXCLUSIVELY WITH (ordering activity):

(1) The following contract items can be ordered under this BPA. All orders placed against this BPA are subject to the terms and conditions of the contract, except as noted below:

MODEL NUMBER/PART NUMBER	*SPECIAL BPA DISCOUNT/PRICE
_____	_____
_____	_____
_____	_____

(2) Delivery:

DESTINATION	DELIVERY SCHEDULES / DATES
_____	_____
_____	_____
_____	_____

(3) The ordering activity estimates, but does not guarantee, that the volume of purchases through this agreement will be _____.

(4) This BPA does not obligate any funds.

(5) This BPA expires on _____ or at the end of the contract period, whichever is earlier.

(6) The following office(s) is hereby authorized to place orders under this BPA:

OFFICE	POINT OF CONTACT
_____	_____
_____	_____
_____	_____

(7) Orders will be placed against this BPA via Electronic Data Interchange (EDI), FAX, or paper.

(8) Unless otherwise agreed to, all deliveries under this BPA must be accompanied by delivery tickets or sales slips that must contain the following information as a minimum:

(a) Name of Contractor;

(b) Contract Number;

(c) BPA Number;

(d) Model Number or National Stock Number (NSN);

(e) Purchase Order Number;

(f) Date of Purchase;

(g) Quantity, Unit Price, and Extension of Each Item (unit prices and extensions need not be shown when incompatible with the use of automated systems; provided, that the invoice is itemized to show the information); and

(h) Date of Shipment.

(9) The requirements of a proper invoice are specified in the Federal Supply Schedule contract. Invoices will be submitted to the address specified within the purchase order transmission issued against this BPA.

(10) The terms and conditions included in this BPA apply to all purchases made pursuant to it. In the event of an inconsistency between the provisions of this BPA and the Contractor's invoice, the provisions of this BPA will take precedence.

****

BASIC GUIDELINES FOR USING "CONTRACTOR TEAM ARRANGEMENTS"

Federal Supply Schedule Contractors may use "Contractor Team Arrangements" (see FAR 9.6) to provide solutions when responding to ordering activity requirements.

These Team Arrangements can be included under a Blanket Purchase Agreement (BPA). BPAs are permitted under all Federal Supply Schedule contracts.

Orders under a Team Arrangement are subject to terms and conditions or the Federal Supply Schedule Contract.

Participation in a Team Arrangement is limited to Federal Supply Schedule Contractors.

Customers should refer to FAR 9.6 for specific details on Team Arrangements.

Here is a general outline on how it works:

- The customer identifies their requirements.
- Federal Supply Schedule Contractors may individually meet the customer's needs, or -
- Federal Supply Schedule Contractors may individually submit a Schedules "Team Solution" to meet the customer's requirement.
- Customers make a best value selection.