Programming, Advanced

6641 36 weeks

Table of Contents

Acknowledgments ......................................................................................................................................... 1
Course Description ........................................................................................................................................ 2
Task Essentials Table .................................................................................................................................... 3
Curriculum Framework .................................................................................................................................. 5
Developing Object-Oriented Programming (OOP) ...................................................................................... 5
Developing Database Applications ............................................................................................................. 14
Developing Interactive Multimedia Applications ....................................................................................... 19
Developing Connected Applications (Mobile and/or Web) ........................................................................ 20
Preparing for Industry Certification ............................................................................................................ 25
Developing Employability Skills ................................................................................................................ 27
SOL Correlation by Task ............................................................................................................................ 31
Cyber Security and Cyber Forensics Infusion Units ................................................................................... 35
Entrepreneurship Infusion Units ................................................................................................................. 35
Microsoft Imagine Academy Resources ..................................................................................................... 35
Appendix: Credentials, Course Sequences, and Career Cluster Information ............................................. 36

Acknowledgments

The components of this instructional framework were developed by the following curriculum development team members:

Marc Bloom, Vice President of Core Infrastructure, Virginia Credit Union, Richmond
Selwyn Lawrence, Instructor, South Lakes High School, Fairfax County Public Schools
Kirsten Poland, Instructor, Washington-Liberty High School, Arlington County Public Schools
Dr. David Raymond, Director, IT Security Office and Lab, Virginia Tech, Blacksburg
Brian Taylor, Instructor, Franklin Military Academy, Richmond City Public Schools
Jeffery Timmerman, Instructor, Brooke Point High School, Stafford County Public Schools
Michael Weber, Instructor, Advanced Technology Center, Virginia Beach City Public
Course Description

Suggested Grade Level: 11 or 12
Prerequisite: 6640

Building on their foundation of programming skills, Advanced Programming students use object-oriented programming to develop database applications, interactive multimedia applications including game applications, mobile applications, and web applications. Students continue to develop their employability skills as they research pathways for continuing education and careers in the information technology industry and engage in various career-building activities.
## Task Essentials Table

- Tasks/competencies designated by plus icons (➕) in the left-hand column(s) are essential
- Tasks/competencies designated by empty-circle icons (_circle) are optional
- Tasks/competencies designated by minus icons (➖) are omitted
- Tasks marked with an asterisk (*) are sensitive.

<table>
<thead>
<tr>
<th>Task Number</th>
<th>6641</th>
<th>Tasks/Competencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing Object-Oriented Programming (OOP)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>📌</td>
<td>Explain the reasoning behind the steps in the software development life cycle (SDLC).</td>
</tr>
<tr>
<td>40</td>
<td>📌</td>
<td>Describe the primary principles of object-oriented programming (OOP).</td>
</tr>
<tr>
<td>41</td>
<td>📌</td>
<td>Identify potential objects, attributes, and methods within a problem description.</td>
</tr>
<tr>
<td>42</td>
<td>📌</td>
<td>Design attributes (properties) and methods of each class within a problem description.</td>
</tr>
<tr>
<td>43</td>
<td>📌</td>
<td>Create a program with user-defined classes.</td>
</tr>
<tr>
<td>44</td>
<td>📌</td>
<td>Describe the concept of reusability.</td>
</tr>
<tr>
<td>45</td>
<td>📌</td>
<td>Identify reusable resources that will help solve a problem.</td>
</tr>
<tr>
<td>46</td>
<td>📌</td>
<td>Import code from existing sources.</td>
</tr>
<tr>
<td>47</td>
<td>📌</td>
<td>Describe the concepts of overloading and overriding methods in an object-oriented language.</td>
</tr>
<tr>
<td>48</td>
<td>📌</td>
<td>Code a program that uses looping structures, conditional structures, and sequential control structures.</td>
</tr>
<tr>
<td>49</td>
<td>📌</td>
<td>Analyze abstract data types.</td>
</tr>
<tr>
<td>50</td>
<td>📌</td>
<td>Implement searching and sorting algorithms.</td>
</tr>
<tr>
<td>51</td>
<td>📌</td>
<td>Code a program that uses error-handling and input-validation procedures.</td>
</tr>
<tr>
<td>52</td>
<td>📌</td>
<td>Code a program to use an interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>53</td>
<td>+</td>
<td>Create a test suite that will verify proper operation of a class or group of related classes.</td>
</tr>
<tr>
<td>54</td>
<td>+</td>
<td>Perform a peer review and test of a program.</td>
</tr>
<tr>
<td>55</td>
<td>+</td>
<td>Analyze code that uses recursion.</td>
</tr>
</tbody>
</table>

**Developing Database Applications**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>+</td>
<td>Identify relational database terminology.</td>
</tr>
<tr>
<td>57</td>
<td>+</td>
<td>Identify database model types.</td>
</tr>
<tr>
<td>58</td>
<td></td>
<td>Describe the three-layer/tier model for database applications.</td>
</tr>
<tr>
<td>59</td>
<td>+</td>
<td>Identify the data object model for the program language.</td>
</tr>
<tr>
<td>60</td>
<td>+</td>
<td>Design a GUI for a database application.</td>
</tr>
<tr>
<td>61</td>
<td>+</td>
<td>Write code to integrate an existing database into a program application.</td>
</tr>
<tr>
<td>62</td>
<td>+</td>
<td>Write code to manage a database.</td>
</tr>
<tr>
<td>63</td>
<td>+</td>
<td>Bind database fields to the interface elements (controls).</td>
</tr>
</tbody>
</table>

**Developing Interactive Multimedia Applications**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>+</td>
<td>Write a design document for a game.</td>
</tr>
<tr>
<td>65</td>
<td>+</td>
<td>Code a multiplayer game.</td>
</tr>
</tbody>
</table>

**Developing Connected Applications (Mobile and/or Web)**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>+</td>
<td>Determine the programming languages used to create connected applications.</td>
</tr>
<tr>
<td>67</td>
<td>+</td>
<td>Design a web application with security features.</td>
</tr>
<tr>
<td>68</td>
<td></td>
<td>Design a GUI for a connected application.</td>
</tr>
<tr>
<td>69</td>
<td></td>
<td>Code a web application (e.g., shopping cart) for a smart device or emulator.</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td>Describe the process of mobile application deployment.</td>
</tr>
<tr>
<td>71</td>
<td></td>
<td>Describe the web application publishing process.</td>
</tr>
<tr>
<td>72</td>
<td>+</td>
<td>Describe client-side and server-side applications.</td>
</tr>
</tbody>
</table>
Preparing for Industry Certification

73  ★   Describe the process and requirements for obtaining industry certifications related to the Programming, Advanced course.

74  ★   Identify testing skills/strategies for a certification examination.

75  ★   Demonstrate ability to successfully complete selected practice examinations (e.g., practice questions similar to those on certification exams).

76  ○   Successfully complete an industry certification examination representative of skills learned in this course (e.g., MCP, IC3).

Developing Employability Skills

77  ★   Investigate continuing education pathways and careers in the information technology industry.

78  ★   Create or update a résumé.

79  ★   Update professional portfolio.

80  ★   Deliver an oral presentation of programming projects.

Legend: ★Essential  ○Non-essential  ☐Omitted

Curriculum Framework

Developing Object-Oriented Programming (OOP)

Task Number 39

Explain the reasoning behind the steps in the software development life cycle (SDLC).

Definition
Explanation should include the reason for each step in the sequence of steps used for program development:

- Define the problem.
- Develop the algorithm.
- Code the program.
- Test the program.
- Debug the program.
- Document the program.
- Implement the program.
- Maintain the program.

Software development can be done in a sequential (Waterfall) or iterative (Agile) method.

**FBLA Competitive Events and Activities Areas**

**Computer Game & Simulation Programming**

**Mobile Application Development**

---

**Task Number 40**

Describe the **primary principles of object-oriented programming (OOP).**

**Definition**

Description should include the following concepts:

- **Abstraction** is a general model of a real-world system that includes the essence of that system.
- **Encapsulation** means that each class should contain all the aspects of its area of responsibility without revealing the details of how it does what it does.
- **Inheritance** allows one class, called a subclass, to use methods and properties of another class, called a superclass.
- **Polymorphism** is the idea that a code element can transform into multiple forms or behave in multiple ways.

**FBLA Competitive Events and Activities Areas**

**Computer Game & Simulation Programming**
Task Number 41

Identify potential objects, attributes, and methods within a problem description.

Definition

Identification should consist of using Unified Modeling Language (UML) and/or use cases for analyzing the problem description to specify the objects, attributes, and methods.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 42

Design attributes (properties) and methods of each class within a problem description.

Definition

Design should include determining the appropriate attributes and methods for each class used in the program to solve the problem description.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 43
Create a program with user-defined classes.

Definition

Creation should use the design-create methods and properties for each class. The program should use the class parameter of the create function statement to assign it to a class, ensuring that the name of the user-defined class agrees with the name assigned in the create function statement.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 44

Describe the concept of reusability.

Definition

Description should include

- making use of existing software libraries created by reputable online sources to help solve aspects of problem descriptions efficiently
- creating libraries to avoid recreating code from scratch, whenever possible
- explaining how abstraction is used in supporting code development.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 45

Identify reusable resources that will help solve a problem.

Definition
Identification should include researching sample resources and references such as the following:

- Online help
- Program documentation
- Program libraries
- Online tutorials
- Developers’ websites
- Published resources
- Application programming interface (API)
- Pre-constructed programs or code

Findings should be properly documented with all necessary attributions to program developers.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

---

**Task Number 46**

**Import code from existing sources.**

**Definition**

Importing should include the code and related libraries and classes from sources such as pre-constructed code, API, and online resources.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

---

**Task Number 47**

**Describe the concepts of overloading and overriding methods in an object-oriented language.**
Definition

Description should include the following concepts:

- **Overloading** is the process of using the same operator symbol or reference for many different functions.
- **Overriding** is the process of re-implementing a method already implemented in a superclass, thereby overriding the inherited property or method.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

---

Task Number 48

**Code a program that uses looping structures, conditional structures, and sequential control structures.**

Definition

Coding should include

- looping structures, such as
  - For/For Each loop
  - While loop
  - Do loop
  - Do Until loop
  - Do While loop
  - Nested loop
- conditional structures, such as
  - If, If/Else, Else-If
  - Switch or Case
- sequential control structures, such as
  - inline
  - functions
  - procedures
  - events
  - methods.

FBLA Competitive Events and Activities Areas
Task Number 49

Analyze abstract data types.

Definition

Analysis may include the following types:

- **Array**: a structure comprised of a group of variables under a single name and accessed by the index
- **Graph**: a structure in which nodes can be connected to any other node
- **Linked-list**: a structure in which each item is linked to the next one by means of a pointer
- **Queue**: a structure accessed by inserting items at one end and removing them from the other
- **Stack**: a dynamic structure accessed from only one end
- **Tree**: a structure begun at the root node in which each node can have child nodes

Task Number 50

Implement searching and sorting algorithms.

Definition

Implementation should make use of the program statements available with the language employed, including

- searching (e.g., sequential/linear or binary) to examine the elements in order according to the implementing technique and proceeding until a match is found
• sorting (e.g., selection, insertion, merge, or bubble) to arrange the components of an array in either ascending or descending order.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

---

**Task Number 51**

**Code a program that uses error-handling and input-validation procedures.**

**Definition**

Coding should include error handling and input validation accomplished by any of the following statements:

- Try/Catch
- Try/Catch/Finally
- If/Then
- Program-defined methods or functions
- Message box validating keyword

---

**Task Number 52**

**Code a program to use an interface.**

**Definition**

Coding should include
• creating one or more interfaces
• providing an implementation (e.g., a concrete class) for one or more interfaces
• demonstrating polymorphism by relying on interfaces rather than concrete implementations.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 53

Create a test suite that will verify proper operation of a class or group of related classes.

Definition

Creation of the test suite should identify normal and corner cases and the appropriate program behavior for each.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 54

Perform a peer review and test of a program.

Definition

Performance should include having team members review and test the program for runtime, syntax, and logic errors, as well as documentation and design problems or bugs.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming
Mobile Application Development

Task Number 55

Analyze code that uses recursion.

Definition

Analysis should include

- reading recursive code
- determining the result
- comparing recursion and looping.

Developing Database Applications

Task Number 56

Identify relational database terminology.

Definition

Identification should include the following terms and their definitions:

- **Relational database**—a database that stores information in one or more tables
- **Table**—a group of related records, with each record in the table pertaining to the same topic and composed of the same fields
- **Record**—a row or entry in the table
- **Field**—a single item or column of information about something (e.g., name, address, or phone number)
- **Data type**—examples include VARCHAR2(), NUMBER(), DATE, CHAR, LONG, BLOB, CLOB
- **Relationships**—types include one-to-one, one-to-many, and many-to-many
- **Primary key**—a special relational database table column (or combination of columns) designated to uniquely identify all table records
• *Structured Query Language (SQL)*—a standard computer language for relational database management and data manipulation used to query, insert, update, and modify data

• *Index*—a method for accessing data in a table

**FBLA Competitive Events and Activities Areas**

Computer Game & Simulation Programming

Database Design & Applications

Mobile Application Development

**Microsoft Imagine Academy Resources**

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6

*Introduction to Database Programs*

---

**Task Number 57**

**Identify database model types.**

**Definition**

Identification should include the following model types:

• **Relational**: the most common type of database; one in which relations exist between tables (e.g., Access, Oracle, MySQL)

• **Hierarchical**: one in which elements are presented in a hierarchy

• **Object-oriented**: the newest type of database; one in which elements are represented by objects

• **Flat file**: the oldest type of database; one in which data are stored in a text file

**FBLA Competitive Events and Activities Areas**

Computer Game & Simulation Programming

Database Design & Applications

Mobile Application Development
Task Number 58

Describe the three-layer/tier model for database applications.

Definition

Description should include the following three layers:

- **Presentation layer**, where users access and interact with the program (the graphical user interface [GUI])
- **Rules layer**, where code is implemented for the rules of processes (e.g., in a banking application: cannot withdraw more money than is in an account); also called the business or application layer
- **Data warehouse layer**, where data are stored within the database

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Database Design & Applications

Mobile Application Development


Task Number 59

Identify the data object model for the program language.

Definition

Identification should include the following concepts:

- The data object model is the data-binding model that exists between a programming language and multiple databases.
- The data object model eliminates the need to write different code for various database platforms.
- The data object model is different for various languages.
- Programming languages correlated to their associated data object models include but are not limited to
  - C#—.Net
  - Java—Java Database Connectivity (JDBC)
Flash ActionScript—Asynchronous JavaScript and XML (AJAX)
Open source languages—PostgreSQL, NoSQL, MySQL.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming
Database Design & Applications
Mobile Application Development

Task Number 60

Design a GUI for a database application.

Definition

Design should include

- the input field
- the output field
- buttons/navigation
- look-and-feel stylistic components (e.g., color scheme, fonts, layout)
- accessibility guidelines.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming
Database Design & Applications
Mobile Application Development

Task Number 61

Write code to integrate an existing database into a program application.

Definition
Coding should make use of programming-language commands and tools to provide data binding between the program application and the database.

**FBLA Competitive Events and Activities Areas**

Computer Game & Simulation Programming

Database Design & Applications

Mobile Application Development

---

**Task Number 62**

**Write code to manage a database.**

**Definition**

Coding should include processes for

- adding records
- updating and editing records
- saving records
- deleting records
- retrieving records based on primary keys
- checking for redundancy
- securing records.

**FBLA Competitive Events and Activities Areas**

Computer Game & Simulation Programming

Database Design & Applications

Mobile Application Development

---

**Task Number 63**

**Bind database fields to the interface elements (controls).**

**Definition**
Binding should include using the tool or wizard associated with the programming language to bind the GUI elements to the database fields (e.g., Visual Basic uses the Data Source Configuration wizard).

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Database Design & Applications

Mobile Application Development

Developing Interactive Multimedia Applications

Task Number 64

Write a design document for a game.

Definition

The document should include

- game title
- game purpose/genre
- target audience
- platform (e.g., PC, iPhone, PlayStation)
- Entertainment Software Rating Board (ESRB) game rating
- game description
- level design (i.e., storyboard)
- characters and objects with behaviors
- instructions for players explaining how to play
- scoring, win/lose conditions, and objectives.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming
Task Number 65

Code a multiplayer game.

Definition

Coding a game that is played by multiple players simultaneously should include

- objects
- behaviors of objects
- environment (i.e., background, audio, colors)
- scoring and win/lose conditions
- instructions for players/end users, with how-to-play and help screens
- credits
- separate key and/or non-key controls development for each player
- the score for each player
- the nature of the competition and defining the method for winning.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Developing Connected Applications (Mobile and/or Web)

Task Number 66

Determine the programming languages used to create connected applications.
Definition

Determination should include, but not be limited to, selecting the appropriate language for

- mobile devices
  - Objective-C and Mac platform—for smart devices, such as iPhone, iPod touch
  - Android, Java, and the Android software development kit (SDK)—for mobile development
- web applications
  - ColdFusion (Adobe platform)
  - Flex (Adobe platform)
  - JSP.NET (Sun/Java platform)
  - Perl (open-source)
  - PHP (open-source)
  - Python (open-source)
  - Ruby (open-source)
  - XHTML.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

---

Task Number 67

Design a web application with security features.

Definition

Design should include

- using Secure Sockets Layer (SSL) protocol to encrypt data communicated with the application
- asserting a user via authentication (e.g., “Who are you?”) that requires a username and password
- protecting restricted resources with authorization (e.g., “What can you do?”), using role-based security
- storing sensitive data (e.g., passwords) via encryption.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming
E-Business

Mobile Application Development

Website Design
The topic for this event changes from year to year. The annual topic may or may not correlate with this particular course. Please refer to the current Virginia FBLA State Handbook.

Task Number 68

Design a GUI for a connected application.

Definition

Design should include

- input field
- output field buttons and navigation
- look-and-feel components (e.g., color scheme, fonts, layout).

Design should be

- optimized for downloading speeds, reduced file size, input and response screens
- modified, based on feedback on layout and design techniques from stakeholders and members of the intended audience
- maintained, based on recommendations and project specifications
- in compliance with accessibility guidelines.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 69

Code a web application (e.g., shopping cart) for a smart device or emulator.
Definition

Coding should include optimization for the constraints of the smart device (e.g., cell phone, tablet) or emulator and should follow best practices for object-oriented design and coding.

Coding should be

- optimized for constraints of the various web browser document object models (e.g., Firefox, Safari, Opera, Internet Explorer, Chrome)
- in compliance with best practices for object-oriented design and coding.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 70

Describe the process of mobile application deployment.

Definition

Description should include

- choosing the smart device through an interface
- following directions for application deployment, based on the device and programming language used.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

Mobile Application Development

Task Number 71

Describe the web application publishing process.

Definition
Description should include

- web server ID and password
- web server FTP/SFTP URL
- software used to publish (e.g., Dreamweaver, Core FTP LE, FireFTP, Expression, FileZilla)
- acquisition of server space
- domain name registration and configuration
- ability to test in a web server environment.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

E-Business

Mobile Application Development

Website Design
The topic for this event changes from year to year. The annual topic may or may not correlate with this particular course. Please refer to the current Virginia FBLA State Handbook.

Task Number 72

Describe client-side and server-side applications.

Definition

Description of the client-side application should include

- how the application communicates with a server application (e.g., a client machine requests a document on a remote web server via a network server, and the network server routes the information back to the client’s browser)
- the appropriate client scripting language (e.g., JavaScript).

Description of the server-side application should include

- how the application is executed on a web server and communicates with the browser
- the appropriate server scripting language (e.g., .NET, PHP, ColdFusion, Flex, Perl, Python, Ruby).

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming
Preparing for Industry Certification

Task Number 73

Describe the process and requirements for obtaining industry certifications related to the Programming, Advanced course.

Definition

The description should include a list of industry certifications related to the Programming, Advanced course and the process/requirements for obtaining the certifications from:

- official websites of the testing organization/vendor
- materials from publishers that have developed practice materials and tests based on information from the testing organization/vendor
- information from certified instructors or industry-certified professionals
- information in the "Course Description" section of this document.

FBLA Competitive Events and Activities Areas

Job Interview

Task Number 74
Identify testing skills/strategies for a certification examination.

Definition

The identification of testing skills and strategies should be undertaken by

- conducting an Internet research project
- reviewing materials from exam and practice-exam publishers
- interviewing certified instructors and/or industry-certified professionals.

FBLA Competitive Events and Activities Areas

Job Interview

Task Number 75

Demonstrate ability to successfully complete selected practice examinations (e.g., practice questions similar to those on certification exams).

Definition

The demonstration should include successfully completing practice examinations for selected certifications related to the course obtained from vendor sites and/or materials from publishers. The level of performance on a practice examination serves as a gauge of the applicant's readiness for formal industry testing.

FBLA Competitive Events and Activities Areas

Job Interview

Task Number 76

Successfully complete an industry certification examination representative of skills learned in this course (e.g., MCP, IC3).
Definition

The successful completion of an industry certification examination will be achieved when the student applicant earns an examination score deemed "passing" by the testing organization. Qualifying examinations are those currently approved at the state level as representative of Programming, Advanced skills.

Students should be encouraged to attain industry certification as evidence of their programming skill level and general employability.

FBLA Competitive Events and Activities Areas

Computer Applications

Database Design & Applications

Spreadsheet Applications

Word Processing

Developing Employability Skills

Task Number 77

Investigate continuing education pathways and careers in the information technology industry.

Definition

Investigation should yield

- a list of postsecondary institutions offering degrees or certificate programs in the following areas of the IT and game-programming industries:
  - Computer science
  - Information technology
  - Game design
  - Game art/graphic design
o Game writing
  o Interdisciplinary studies
- a list of job titles related to each field
- the preparation required for each job
- the opportunities for advancement
- employment trends in each field.

Job databanks should also be used to help students match their interests, abilities, aptitudes, and job expectations to industry standards.

**FBLA Competitive Events and Activities Areas**

**Computer Game & Simulation Programming**

**E-Business**

**Introduction to Information Technology**

**Mobile Application Development**

**Website Design**
The topic for this event changes from year to year. The annual topic may or may not correlate with this particular course. Please refer to the current Virginia FBLA State Handbook.

**Microsoft Imagine Academy Resources**

Digital Technology and Career Opportunities

---

**Task Number 78**

**Create or update a résumé.**

**Definition**

The résumé should be created or updated using an appropriate software application and should include

- a technical skills section
- educational background
- work history
- honors and awards
- memberships in student, professional, and/or community organizations, with activities completed, leadership positions held, and community service performed
- a separate cover letter and reference page
- evidence that cautions regarding the posting of personal information online have been heeded
- accessibility in multiple file formats, including PDF.

**FBLA Competitive Events and Activities Areas**

**Computer Game & Simulation Programming**

**E-Business**

**Electronic Career Portfolio**

**Introduction to Information Technology**

**Job Interview**

**Mobile Application Development**

**Website Design**

The topic for this event changes from year to year. The annual topic may or may not correlate with this particular course. Please refer to the current Virginia FBLA State Handbook.

---

**Task Number 79**

**Update professional portfolio.**

**Definition**

Portfolio should include

- digital and non-digital documents (e.g., program design, source code, technical documentation, output) that form a representative sample the student’s qualifications, knowledge, experience, skills, and abilities
- a résumé in digital and traditional formats.

**FBLA Competitive Events and Activities Areas**

**Computer Game & Simulation Programming**

**E-Business**

**Electronic Career Portfolio**
Introduction to Information Technology

Job Interview

Mobile Application Development

Website Design
The topic for this event changes from year to year. The annual topic may or may not correlate with this particular course. Please refer to the current Virginia FBLA State Handbook.

Task Number 80

Deliver an oral presentation of programming projects.

Definition

Delivery should include

- displaying evidence of thorough preparation and knowledge of the material
- providing relevant visual aids
- maintaining eye contact with the audience as much as possible
- speaking clearly and distinctly with confidence and enthusiasm
- demonstrating professional presentation skills, such as standing up straight and avoiding excessive movement and other nervous habits
- responding to questions in an effective manner.

FBLA Competitive Events and Activities Areas

Computer Game & Simulation Programming

E-Business

Electronic Career Portfolio

Introduction to Information Technology

Mobile Application Development

Website Design
The topic for this event changes from year to year. The annual topic may or may not correlate with this particular course. Please refer to the current Virginia FBLA State Handbook.
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>English:</th>
<th>Mathematics:</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>Explain the reasoning behind the steps in the software development life cycle (SDLC).</td>
<td>11.5, 12.5</td>
<td>COM.1, COM.2, COM.3, COM.4, COM.18</td>
</tr>
<tr>
<td>40</td>
<td>Describe the primary principles of object-oriented programming (OOP).</td>
<td>11.5, 12.5</td>
<td>COM.1</td>
</tr>
<tr>
<td>41</td>
<td>Identify potential objects, attributes, and methods within a problem description.</td>
<td>11.5, 12.5</td>
<td>COM.1, COM.5</td>
</tr>
<tr>
<td>42</td>
<td>Design attributes (properties) and methods of each class within a problem description.</td>
<td>11.1, 12.1</td>
<td>COM.1, COM.3, COM.4, COM.5, COM.10, COM.11</td>
</tr>
<tr>
<td>43</td>
<td>Create a program with user-defined classes.</td>
<td>11.5, 12.5</td>
<td>COM.1, COM.2, COM.5, COM.6, COM.15</td>
</tr>
<tr>
<td>44</td>
<td>Describe the concept of reusability.</td>
<td>11.5, 12.5</td>
<td>COM.1, COM.5, COM.7, COM.9</td>
</tr>
<tr>
<td>45</td>
<td>Identify reusable resources that will help solve a problem.</td>
<td>11.5, 12.5</td>
<td>COM.1, COM.7, COM.9</td>
</tr>
<tr>
<td>46</td>
<td>Import code from existing sources.</td>
<td>11.2, 12.2</td>
<td>COM.1, COM.7, COM.9</td>
</tr>
<tr>
<td>47</td>
<td>Describe the concepts of overloading and overriding methods in an object-oriented language.</td>
<td>11.5, 12.5</td>
<td>COM.1</td>
</tr>
<tr>
<td>48</td>
<td>Code a program that uses looping structures, conditional structures, and sequential control structures.</td>
<td>11.5, 12.5</td>
<td>COM.1, COM.8, COM.13, COM.14, DM.9*</td>
</tr>
<tr>
<td>49</td>
<td>Analyze abstract data types.</td>
<td>11.5, 12.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exercise</td>
<td>Subject Areas</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Implement searching and sorting algorithms.</td>
<td>Mathematics: COM.1, COM.14, DM.4, DM.3*</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Code a program that uses error-handling and input-validation procedures.</td>
<td>Mathematics: COM.1, COM.2, COM.3, COM.10, COM.17, COM.18</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>Code a program to use an interface.</td>
<td>Mathematics: AII.3, COM.1, COM.2</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>Create a test suite that will verify proper operation of a class or group of related classes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>Perform a peer review and test of a program.</td>
<td>English: 11.1, 12.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.1, COM.2, COM.17, COM.18</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>Analyze code that uses recursion.</td>
<td>English: 11.5, 12.5</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>Identify relational database terminology.</td>
<td>English: 11.5, 12.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.1, COM.9, COM.14, COM.15</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>Identify database model types.</td>
<td>English: 11.5, 12.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.1, COM.9, COM.14</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>Describe the three-layer/tier model for database applications.</td>
<td>English: 11.5, 12.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.1</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>Identify the data object model for the program language.</td>
<td>English: 11.5, 12.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.1, COM.2</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>Design a GUI for a database application.</td>
<td>English: 11.5, 12.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.1, COM.2, COM.10, COM.11, COM.12</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>Write code to integrate an existing database into a program application.</td>
<td>Mathematics: COM.1, COM.6, COM.7, COM.15</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>Write code to manage a database.</td>
<td>Mathematics: COM.1, COM.2, COM.7, COM.8, COM.9, COM.13, COM.14</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>Bind database fields to the interface elements (controls).</td>
<td>Mathematics: COM.1, COM.7</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>Write a design document for a game.</td>
<td>English: 11.6, 11.7, 12.6, 12.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.1, COM.2, COM.3, COM.4, COM.5, COM.10, COM.11</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Objective</strong></td>
<td><strong>Mathematics</strong></td>
<td><strong>English</strong></td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>66</td>
<td>Determine the programming languages used to create connected applications.</td>
<td>AFDA.1, AII.3, COM.1, COM.2, COM.4, COM.6, COM.10, COM.11, COM.12, COM.15</td>
<td>11.5, 11.8, 12.5, 12.8</td>
</tr>
<tr>
<td>67</td>
<td>Design a web application with security features.</td>
<td>COM.1, COM.2, COM.4, COM.5, COM.6, COM.10, COM.11, COM.15</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>Design a GUI for a connected application.</td>
<td>COM.1, COM.2, COM.4, COM.5, COM.6, COM.10, COM.11, COM.15</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>Code a web application (e.g., shopping cart) for a smart device or emulator.</td>
<td>COM.1, COM.2</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>Describe the process of mobile application deployment.</td>
<td></td>
<td>11.5, 12.5</td>
</tr>
<tr>
<td>71</td>
<td>Describe the web application publishing process.</td>
<td></td>
<td>11.5, 12.5</td>
</tr>
<tr>
<td>72</td>
<td>Describe client-side and server-side applications.</td>
<td></td>
<td>11.5, 12.5</td>
</tr>
<tr>
<td>73</td>
<td>Describe the process and requirements for obtaining industry certifications</td>
<td></td>
<td>11.5, 11.8, 12.5, 12.8</td>
</tr>
<tr>
<td></td>
<td>related to the Programming, Advanced course.</td>
<td></td>
<td>History and Social Science: GOVT.7, GOVT.8, GOVT.14, GOVT.15</td>
</tr>
<tr>
<td>74</td>
<td>Identify testing skills/strategies for a certification examination.</td>
<td></td>
<td>11.5, 12.5</td>
</tr>
<tr>
<td>75</td>
<td>Demonstrate ability to successfully complete selected practice examinations</td>
<td></td>
<td>11.5, 12.5</td>
</tr>
<tr>
<td></td>
<td>(e.g., practice questions similar to those on certification exams).</td>
<td></td>
<td>History and Social Science: GOVT.7, GOVT.8, GOVT.14, GOVT.15</td>
</tr>
<tr>
<td>76</td>
<td>Successfully complete an industry certification examination representative of</td>
<td></td>
<td>11.5, 12.5</td>
</tr>
<tr>
<td></td>
<td>skills learned in this course (e.g., MCP, IC3).</td>
<td></td>
<td>History and Social Science: GOVT.7, GOVT.8, GOVT.14, GOVT.15</td>
</tr>
<tr>
<td>77</td>
<td>Investigate continuing education pathways and careers in the information</td>
<td></td>
<td>11.5, 11.8, 12.5, 12.8</td>
</tr>
<tr>
<td></td>
<td>technology industry.</td>
<td></td>
<td>History and Social Science: GOVT.7, GOVT.8, GOVT.14, GOVT.15</td>
</tr>
</tbody>
</table>
Instructional Scenarios

The following instructional scenarios provide classroom activities to support the major concepts included in Advanced Programming.

- **Cafeteria Catastrophe**
  Duty/Concept Areas: Developing Object-Oriented Programming; Developing Database Applications; Developing Connected Applications (Mobile and/or Web); Developing Employability Skills
  Students will create a program to track cafeteria inventory and sales, calculate purchases, and store student records.

- **Fantasy Finale**
  Duty/Concept Area: Developing Object-Oriented Programming
  Students will develop video-game characters and build a program to test them together in the same environment.

- **An App to Track FBLA Committees**
  Duty/Concept Areas: Developing Object-Oriented Programming; Developing Interactive Multimedia Applications; Developing Connected Applications (Mobile and/or Web)
  Students will develop an app in which users can add and remove themselves from FBLA committees.

- **Putting It All Together**
  Duty/Concept Areas: Developing Object-Oriented Programming; Developing Database Applications; Developing Web Applications; Developing Connected Applications (Mobile and/or Web)
  Students will select a team of programmers and bid to win a contract to develop applications for a sales business.

- **The SOL Game**
  Duty/Concept Areas: Developing Object-Oriented Programming; Developing Interactive Multimedia Applications
  Students will develop a game to be used for reviewing basic mathematics skills.

- **Video Game Rental**
  Duty/Concept Areas: Developing Object-Oriented Programming; Developing Database Applications
  Students will develop an online interface through which customers can search, select, and check out games.

- **You Be the Judge**
  Duty/Concept Areas: Developing Connected Applications (Mobile and/or Web); Developing Employability Skills
  Students will investigate a case in which an employee misrepresents credentials and allows company software to be compromised.
Cyber Security and Cyber Forensics Infusion Units

Cyber Security and Cyber Forensics Infusion Units (CYBR) were designed to be infused with designated CTE courses to help students achieve additional, focused, validated tasks/competencies in personal and professional cyber security skills. These units are not mandatory, and, as such, the tasks/competencies are marked as “optional” and are to be taught at the instructor’s discretion.

Entrepreneurship Infusion Units

Entrepreneurship Infusion Units may be used to help students achieve additional, focused competencies and enhance the validated tasks/competencies related to identifying and starting a new business venture. Because the unit is a complement to certain designated courses and is not mandatory, all tasks/competencies are marked “optional.”

Microsoft Imagine Academy Resources

Microsoft Imagine Academy (MSIA) offers classroom resources and materials and instructional techniques that will help enhance instruction and learning for this course. Using the school’s membership ID and product key for the Microsoft Imagine Academy, all resources are available through the MSIA Member Dashboard on the Microsoft site.

- To access the curriculum resources, select the Classroom Tile from the member site.
- To access downloadable curriculum resources including the MOAC e-Book, Lesson Plans, and Study Guides select Curriculum Overview - Curriculum Downloads.
- To access Online Learning videos and tutorials select Online Learning Directory tile.
- For more information visit: How to Get Started with Microsoft Imagine Academy Program.
Appendix: Credentials, Course Sequences, and Career Cluster Information

Industry Credentials: Only apply to 36-week courses

- Advanced Placement (AP) Computer Science Principles Examination
- App Development with Swift Level 1 Examination
- Certified Associate in Python Programming (PCAP) Examination
- Certified Entry-Level Python Programmer (PCEP) Examination
- Certified Internet Web (CIW) Advanced HTML 5 and CSS 3 Specialist Examination
- Certified Internet Web (CIW) Data Analyst Examination
- Certified Internet Web (CIW) Database Design Specialist Examination
- Certified Internet Web (CIW) E-Commerce Services Specialist Examination
- Certified Internet Web (CIW) Internet Business Associate Examination
- Certified Internet Web (CIW) JavaScript Specialist Examination
- Certified Internet Web (CIW) Site Development Associate Examination
- Certified Internet Web (CIW) Social Media Strategist Examination
- Certified Internet Web (CIW) User Interface Designer Examination
- Certified Internet Web (CIW) Web Design Specialist Examination
- Certified Internet Web (CIW) Web Security Specialist Examination
- Cloud Essentials Certification Examination
- College and Work Readiness Assessment (CWRA+)
- Computer Programming Assessment
- IC3 Digital Literacy Certification Examination
- Microsoft Technology Associate (MTA) Examinations
- National Career Readiness Certificate Assessment
- Unity Certified User Examination
- Workplace Readiness Skills for the Commonwealth Examination

Concentration sequences: A combination of this course and those below, equivalent to two 36-week courses, is a concentration sequence. Students wishing to complete a specialization may take additional courses based on their career pathways. A program completer is a student who has met the requirements for a CTE concentration sequence and all other requirements for high school graduation or an approved alternative education program.

- AP Computer Science A (3185/36 weeks)
- AP Computer Science Principles (10160/36 weeks)
- Computer Information Systems (6612/36 weeks)
- Computer Information Systems (6614/18 weeks)
- Computer Information Systems, Advanced (6613/36 weeks)
- Computer Information Systems, Advanced (6615/18 weeks)
- Computer Network Software Operations (6650/36 weeks)
- Computer Network Software Operations, Advanced (6651/36 weeks)
- Cybersecurity Software Operations (6304/36 weeks)
- Database Design and Management (Oracle) (6660/36 weeks)
- Design, Multimedia, and Web Technologies (6630/36 weeks)
- Design, Multimedia, and Web Technologies (6632/18 weeks)
- Design, Multimedia, and Web Technologies, Advanced (6631/36 weeks)
- Design, Multimedia, and Web Technologies, Advanced (6633/18 weeks)
- Digital Applications (6611/36 weeks)
- Digital Applications (6617/18 weeks)
- Information Technology Fundamentals (6670/36 weeks)
- International Baccalaureate Computer Science (10159/36 weeks)
- International Baccalaureate Information Technology in a Global Society (IB6613/36 weeks)
- Java Programming (Oracle) (6661/36 weeks)
- Modeling and Simulation Technology (8460/36 weeks)
- Office Administration (6621/36 weeks)
- Office Administration (6622/18 weeks)
- Programming (6640/36 weeks)

### Career Cluster: Arts, Audio/Video Technology and Communications

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Occupations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telecommunications</td>
<td>Computer Programmer</td>
</tr>
<tr>
<td></td>
<td>Network Systems and Data Communication Analyst</td>
</tr>
</tbody>
</table>

### Career Cluster: Information Technology

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Occupations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Systems</td>
<td>Computer Security Specialist, Computer Software Engineer, Database Analyst,</td>
</tr>
<tr>
<td></td>
<td>Network Architect, Systems Analyst</td>
</tr>
<tr>
<td>Programming and Software</td>
<td>Applications Integrator, Computer Software Engineer, Game Designer,</td>
</tr>
<tr>
<td>Development</td>
<td>Programmer, Multimedia Artist, Animator, Network Systems and Data</td>
</tr>
<tr>
<td></td>
<td>Communication Analyst, Programmer</td>
</tr>
<tr>
<td></td>
<td>Project Manager, Software Applications Engineer, Software Test Engineer,</td>
</tr>
<tr>
<td></td>
<td>Systems Analyst, Web Developer</td>
</tr>
<tr>
<td>Web and Digital Communications</td>
<td>Applications Integrator, Graphic Designer, Multimedia Artist, Animator,</td>
</tr>
<tr>
<td></td>
<td>Project Manager, Systems Analyst</td>
</tr>
<tr>
<td>Pathway</td>
<td>Occupations</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><strong>Engineering and Technology</strong></td>
<td>Computer Hardware Engineer</td>
</tr>
<tr>
<td></td>
<td>Computer Programmer</td>
</tr>
<tr>
<td></td>
<td>Computer Software Engineer</td>
</tr>
<tr>
<td></td>
<td>Mechanical Engineering Technician</td>
</tr>
<tr>
<td></td>
<td>Network and Computer Systems Administrator</td>
</tr>
<tr>
<td></td>
<td>Network Systems and Data Communication Analyst</td>
</tr>
<tr>
<td></td>
<td>Systems Analyst</td>
</tr>
<tr>
<td></td>
<td>Technical Writer</td>
</tr>
<tr>
<td></td>
<td>Telecommunications Specialist</td>
</tr>
<tr>
<td><strong>Science and Mathematics</strong></td>
<td>Bioinformatics Technician</td>
</tr>
<tr>
<td></td>
<td>Secondary School Teacher</td>
</tr>
<tr>
<td></td>
<td>Technical Writer</td>
</tr>
</tbody>
</table>