Visual C++

Grade 10, 11, or 12

Prerequisite: Visual Basic

Credit Value: 5

ABSTRACT

Visual C++ is designed to introduce students to another programming language in addition to Visual Basic. The language used in this course is a subset of C++. Students are given an introduction to object-oriented programming, briefly touching on inheritance. This course facilitates the design of programs based on modern concepts of an “object-based” approach to programming emphasizing data abstraction, encapsulation, and information hiding. The software used in this course is Visual C++ .NET, which is bundled in Visual Studio.Net. This provides the integrated development environment to enter, compile, build, and debug programs in one package. Students who are interested in a career in computer programming may also enroll in Advanced Placement Computer Programming. Benchmark assessments are employed to track individual student progress.

Adopted by the Somerville Board of Education on November 20, 2012
<table>
<thead>
<tr>
<th>Month/Marking Period</th>
<th>September</th>
<th>October</th>
<th>November</th>
<th>December</th>
<th>January</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essential Question:</td>
<td>What is object-orientated programming language?</td>
<td>How are variables, constants, and scope utilized in programming?</td>
<td>How are logic limitations applied to programming?</td>
<td>How can long code be divided for easier reading?</td>
<td>How are loop structures utilized to control program flow?</td>
</tr>
<tr>
<td>Content:</td>
<td>Introducing Visual C++ Variables and Constants</td>
<td>Application of Logic to Program Structure</td>
<td>Controlling Program Flow with Decision Structures</td>
<td>Controlling Program Flow with Looping Structures</td>
<td></td>
</tr>
<tr>
<td>Skills and Topics:</td>
<td>• review the history of computing</td>
<td>• assign meaningful variable names, choosing the correct data type and steps in the creation of software</td>
<td>• define and use functions and parameters</td>
<td>• execute functions manually</td>
<td>• use string functions</td>
</tr>
<tr>
<td></td>
<td>• discuss computer architecture, number systems, ASCII code, OS, and languages</td>
<td>• use variables, named constants, data-types, cin, expressions, and operators</td>
<td>• define overloaded functions</td>
<td>• name the function and execute the program</td>
<td>• include Pause() in utility programs</td>
</tr>
<tr>
<td></td>
<td>• convert binary to decimal and decimal to binary</td>
<td>• express the rules of C++ identifiers</td>
<td>• differentiate between value and reference parameters</td>
<td>• describe how the program used classes and objects</td>
<td>• learn #include“Z:\common\lvp\pegclass.h”</td>
</tr>
<tr>
<td></td>
<td>• explore converting octal and hex</td>
<td>• convert the built in data types, such as double and float, int and long, and char</td>
<td>• program main functions</td>
<td>• define constructors and object parameters</td>
<td>• learn #include“Z:\utility.h”</td>
</tr>
<tr>
<td></td>
<td>• create a conversion worksheet listing five base 10 numbers (each number must be greater than 20) and five binary numbers</td>
<td>• review integer and modulus division and printing with decimals</td>
<td>• build and document functions, including function prototypes</td>
<td>• use string functions</td>
<td>• write the function for TotalRings()</td>
</tr>
<tr>
<td></td>
<td>• explain converting each base 10 number and binary number from decimal to binary or binary to decimal</td>
<td>• utilize textbox objects to get input from the user</td>
<td>• create a utility library containing Signature(), Pause(), getline(), Drawbar(), and Display Formatted(); complete with function description and pre- and post-conditions</td>
<td>• generate random numbers</td>
<td>• create and modify Towers of Hanoi program challenge</td>
</tr>
<tr>
<td></td>
<td>• discuss ethical responsibilities of the computer programmer along with social and ethical consequences</td>
<td>• apply automatic type conversion and special division operators</td>
<td>• execute functions manually</td>
<td>• return the integer portion of a number without rounding and determine variable lifetime</td>
<td>• identify the pattern of solving the next iteration of the puzzle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• use And, Or, and Not in Boolean expressions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• describe class member functions and class objects in Peg Class</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• apply the do loop structure to programs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• apply the do loop and control infinite loops</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• utilize input boxes in applications</td>
</tr>
</tbody>
</table>
**Visual C++**  
**Grade 10, 11, or 12**

<table>
<thead>
<tr>
<th>Month/Marking Period</th>
<th>September</th>
<th>October</th>
<th>November</th>
<th>December</th>
<th>January</th>
</tr>
</thead>
</table>
| **Skills and Topics:** | • explore rights to privacy, legal issues, and General Moral Imperatives  
• run a simple program and explain the steps of compiling, linking, and executing  
• correct syntax errors and errors that result in warning messages  
• use Visual C++ to key-in the first program, "Hello World"  
• create directories  
• use special characters and program style  
• use cout to display strings and numeric calculations  
• use special characters for display and control of output | • identify and mediate syntax, logic, and runtime errors using the visual basic debugger  
• apply scope, specification, design, coding, testing, and debugging to computer programming | • describe the steps in the creation of software, including specification, design, coding, debugging, and testing  
• develop methodologies for building functions  
• use top-down design in constructing large programs  
• evaluate a program that uses function prototypes in its organization | • construct if statements avoiding common errors  
• add an else clause to the if statement  
• construct compound and nested if statements  
• construct an else-if ladder to decide among three or more actions  
• use while loops  
• apply select case in programming structure | |

**Integration of Technology:**  
- [http://www.asciitable.com](http://www.asciitable.com)  
- [http://www.is.wayne.edu/olmt/binary](http://www.is.wayne.edu/olmt/binary)  
- [http://www.acm.org/about/code-of-ethics](http://www.acm.org/about/code-of-ethics)  
- [http://cpsr.org/issues/ethics/cei/](http://cpsr.org/issues/ethics/cei/)  
- [www.mazeworks.com/hanoi/](http://www.mazeworks.com/hanoi/)  
- Internet, Web Quests, wireless laptop computers, computer laboratory, SMART Boards, Visual Basic software, VoiceThread, Google Docs, email, MacBooks, iPads, video streaming, podcasting

**Writing:**  
Open-ended responses, conclusions and analysis of exploratory activities

**Formative Assessments:**  
Teacher observation, class participation, Do Now activities, opening activities, closing activities, programming code, authentic benchmark assessments

**Summative Assessments:**  
Quizzes, tests, programs, projects, presentations, benchmark assessments

**Performance Assessments:**  
Authentic assessments, programs, projects
### Interdisciplinary Connections:

*ELA: SL.1-6, L.1-6, RST.1-10, WHST.1-2, WHST.4-10  
Arts: Design elements are exemplified in the construction of creating classes.  
World Language: 7.1.AL.A.3  

### 21st Century Themes:

- Global Awareness
- Financial, Economic, Business, and Entrepreneurial Literacy
- Health Literacy

### 21st Century Skills:

- Creativity and Innovation
- Media Literacy
- Critical Thinking and Problem Solving
- Life and Career Skills
- Information and Communication Technologies Literacy
- Communication and Collaboration
- Information Literacy

### Resources:


### Careers:

Applicable career options are discussed as they arise throughout the technology program. Career options include, but are not limited to, the following career clusters: Arts, A/V Technology, and Communications Career Cluster; Business, Management, and Administration Career Cluster; Education and Training Career Cluster; Government and Public Administration Career Cluster; Health Science Career Cluster; Hospitality and Tourism Career Cluster; Human Services Career Cluster; Information Technology Career Cluster; Law, Public Safety, Correction, and Security Career Cluster; Manufacturing Career Cluster; Marketing Career Cluster; Science, Technology, Engineering and Mathematics Career Cluster; Transportation, Distribution, and Logistics Career Cluster.

*2010 Common Core Content Standards:*

- **RL**: Reading Literature  
- **RI**: Reading Informational Text  
- **W**: Writing  
- **SL**: Speaking and Listening  
- **L**: Language  
- **N**: Real Number System  
- **A**: Algebra  
- **F**: Functions  
- **G**: Geometry  
- **S**: Statistics and Probability  
- **MD**: Measurement and Data
### Visual C++
**Grade 10, 11, or 12**

<table>
<thead>
<tr>
<th>Month/Marking Period</th>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
</tr>
</thead>
</table>

#### Essential Question:
- How can we ensure the smooth running of a program?
- How are classes programmed?
- How are algorithms used to allow for recursion?
- How are array structures applicable to the formation of classes in object-oriented programming?
- How does inheritance apply to programming essentials?

#### Content:
- **Basic Debugging Techniques**
- **Using Classes and Objects**
- **Mathematics and Recursion**
- **Arrays and Structs**
- **Streams and Files**

#### Skills and Topics:
- Use sentinels to control loops
- Evaluate a counting and summing loop
- Use `++`, `--`, `+=`, and `-=` operators
- Use the `break` statement to terminate the current block of code
- Distinguish between syntax errors, logic errors, and run time errors
- Interpret build messages – compiler errors and warnings
- Trace application errors with output statements
- Use comments to locate bugs
- Analyze a program to locate logic errors
- Explain and apply the concept of encapsulation
- Create a class constructor member function
- Explain classes and objects, including the concepts of instantiation and member function
- Create header and implementation files to be added to the library
- Use classes in programs
- Use string class member functions
- Define constructors
- Create a class declaration that includes public member functions and private data member declarations
- Use the mathematical function contained in the library, such as trigonometric, logarithmic, and exponential
- Use a tolerance when comparing non-integer values to avoid rounding errors
- Round a double value to the nearest integer
- Use a simple recursive call in a program
- Use the `ifstream` class in the fstream library, including `getline()`, `fail()`, and `>>` operator
- Read numeric and character data from a file
- Use dynamic arrays
- Use a `typedef` statement to associate an identifier with a type
- Maniculate a string object as an array of chars
- Employ arrays with meaningful indexes
- Use two-dimensional arrays to represent a matrix
- Explain how a stream processes characters
- Explain the concept of inheritance
- Use the `fstream` class in the fstream library, including `getline()`, `fail()`, and `>>` operator
- Use `remove()` and `rename()` functions to manipulate files
- Differentiate between sequential and random access files
# Visual C++
## Grade 10, 11, or 12

<table>
<thead>
<tr>
<th>Month/Marking Period</th>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skills and Topics:</td>
<td>• trace program execution with the step commands and breakpoints&lt;br&gt;• utilize input boxes in applications&lt;br&gt;• apply string class in programming code&lt;br&gt;• define string concatenation and use string comparison in code&lt;br&gt;• apply the char structure and use it in code&lt;br&gt;• monitor variables during execution&lt;br&gt;• diagnose a product and system that is malfunctioning using troubleshooting methods</td>
<td>• create a utility class and operator member function&lt;br&gt;• pass an object as a parameter&lt;br&gt;• write a program to solve the towers of Hanoi puzzle&lt;br&gt;• use compiler directives and apply automatically-created member functions&lt;br&gt;• explain the concepts of object-oriented programming and object-oriented design</td>
<td>• implement merge sort and binary search algorithms as recursive functions&lt;br&gt;• implement the depth-first search algorithm</td>
<td>• use the struct statement to create a user-defined type&lt;br&gt;• create a struct constructor&lt;br&gt;• use constants as struct data members&lt;br&gt;• examine an array of structs</td>
<td>• use seekop() and seekg() to manipulate the get and put pointers&lt;br&gt;• use tellg() and tellp() to determine the current positions of the get and put pointers&lt;br&gt;• append data to a file using ios::end&lt;br&gt;• create a file stream that can be read or written to</td>
</tr>
</tbody>
</table>

**Integration of Technology:**

**Writing:**
- Open-ended responses, conclusions and analysis of exploratory activities

**Formative Assessments:**
- Teacher observation, class participation, Do Now activities, opening activities, closing activities, programming code, authentic benchmark assessments

**Summative Assessments:**
- Quizzes, tests, programs, projects, presentations, benchmark assessments

**Performance Assessments:**
- Authentic assessments, programs, projects
# Visual C++

**Grade 10, 11, or 12**

<table>
<thead>
<tr>
<th>Month/Marking Period</th>
<th>February</th>
<th>March</th>
<th>April</th>
<th>May</th>
<th>June</th>
</tr>
</thead>
<tbody>
<tr>
<td>21st Century Themes:</td>
<td>☒ Global Awareness</td>
<td>☐ Civic Literacy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>☒ Financial, Economic, Business, and Entrepreneurial Literacy</td>
<td>☐ Health Literacy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21st Century Skills:</td>
<td>☒ Creativity and Innovation</td>
<td>☒ Media Literacy</td>
<td>☒ Critical Thinking and Problem Solving</td>
<td>☒ Life and Career Skills</td>
<td></td>
</tr>
<tr>
<td></td>
<td>☒ Information and Communication Technologies Literacy</td>
<td>☒ Communication and Collaboration</td>
<td>☒ Information Literacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Careers:</td>
<td>Applicable career options are discussed as they arise throughout the technology program. Career options include, but are not limited to, the following career clusters: Arts, A/V Technology, and Communications Career Cluster; Business, Management, and Administration Career Cluster; Education and Training Career Cluster; Government and Public Administration Career Cluster; Health Science Career Cluster; Hospitality and Tourism Career Cluster; Human Services Career Cluster; Information Technology Career Cluster; Law, Public Safety, Correction, and Security Career Cluster; Manufacturing Career Cluster; Marketing Career Cluster; Science, Technology, Engineering and Mathematics Career Cluster; Transportation, Distribution, and Logistics Career Cluster.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*2010 Common Core Content Standards:*

- RL: Reading Literature
- RI: Reading Informational Text
- W: Writing
- SL: Speaking and Listening
- L: Language
- N: Real Number System
- A: Algebra
- F: Functions
- G: Geometry
- S: Statistics and Probability
- MD: Measurement and Data
Visual C++
Course Requirements

Grade: 10, 11, or 12  Prerequisite: Visual Basic  Credit Value: 5
Length of Course: Academic Year

Course Description

Visual C++ is designed to introduce students to another programming language in addition to Visual Basic. The language used in this course is a subset of C++. Students are given an introduction to object-oriented programming, briefly touching on inheritance. This course facilitates the design of programs based on modern concepts of an “object-based” approach to programming emphasizing data abstraction, encapsulation, and information hiding. The software used in this course is Visual C++.NET, which is bundled in Visual Studio.Net. This provides the integrated development environment to enter, compile, build, and debug programs in one package. Students who are interested in a career in computer programming may also enroll in Advanced Placement Computer Programming. Benchmark assessments are employed to track individual student progress.

Course Content

This course will consist of the following units of study:
- Introducing Visual C++
- Variables and Constants
- Application of Logic to Program Structure
- Controlling Program Flow with Decision Structures
- Controlling Program Flow with Looping Structures
- Basic Debugging Techniques
- Using Classes and Objects
- Mathematics and Recursion
- Arrays and Structs
- Streams and Files

Course Objectives

The student will demonstrate the ability to answer in detail the following essential questions:
- What is object-oriented programming language?
- How are variables, constants, and scope utilized in programming?
- How are logic limitations applied to programming?
- How can long code be divided for easier reading?
Course Objectives (continued)

- How are loop structures utilized to control program flow?
- How can we ensure the smooth running of a program?
- How are classes programmed?
- How are algorithms used to allow for recursion?
- How are array structures applicable to the formation of classes in object-oriented programming?
- How does inheritance apply to programming essentials?
- What are the post-graduation and/or career options that apply to the course content?

Evaluation Process

A final average of 65% or better is required to be awarded course credit. Throughout the length of this course, students may be evaluated on the basis of, but not limited to:

- Formative Assessments, such as writing prompts, journals, and portfolios
- Summative Assessments, such as quizzes, tests, and midterm and final examinations
- Performance Assessments, such as projects and presentations
- Technology-based Applications, such as electronic portfolios, Web Quests, ThinkQuest, and podcasting
- Class Participation
- Homework

Specific weights will be determined by course and level.
Visual C++
Student Agreement

STUDENT NAME: ____________________________________________________________

Last Name          First Name

GRADE: ______________

My signature below indicates that I have received a copy of the
Somerville Public Schools Course Requirements for Visual C++.

I acknowledge my responsibility to read and understand all of the
information contained in the Visual C++ Course Requirements
information and syllabus packet.

___________________________________________  _______________________
Student Signature                  Date

Note: Please share the course requirements for Visual C++ with your parents.