

**East Penn School District
Secondary Curriculum**

**A Planned Course Statement
for
C++ Programming**

Course #369

Grade(s) 9-12

Department: Computer Science

Length of Period (mins.) 42

Total Clock Hours 120

Periods per Week 5

Length of Course (yrs.) 1

Type of Offering: required x elective

CREDIT 1

Adopted 04/27/09

**Developed by:
Carlen Blackstone
Beth Stoudt**

Description of Course #369

Course Title: C++ Programming

Description: This course will provide students with a solid background in structured programming techniques and an introduction to object-oriented programming. Data types, loop structures, decision making, user-defined functions, parameter passing, file input and output, string processing, arrays, and advanced data structures will all be covered. Programming assignments will relate to a large variety of real-life applications.

Goals:

- To enable students to write programs that incorporate good structure, user-oriented interfaces, and sound problem solving strategies.
- To introduce students to all aspects of the C++ programming language in order for them to create fun, engaging, and worthwhile products that are best implemented through computer software.

Requirements:

Prerequisite: Programming Foundations (recommended 80% or better)

Text:

Lambert, Kenneth A. and Nance, Douglas W., **Fundamentals of C++, Understanding Programming and Problem Solving**, Southwestern Educational Publishing, 1998.

Key to Levels of Achievement (listed with each learning objective)

Awareness (A)	Students are introduced to concepts, forms, and patterns.
Learning (L)	Students are involved in a sequence of steps and practice activities which involve further development and allow for evaluation of process.
Understanding (U)	Students demonstrate ability to apply acquired concepts and skills to individual assignments and projects on an independent level.
Reinforcement (R)	Students maintain and broaden understanding of concepts and skills to accomplish tasks at a greater level of sophistication.

Unit	Num	Objective	Level	Content	Evaluation	Standard
I. Computer Science Concepts	1	Students will identify the various aspects of computer science and know what it means to study the field of computer science.	L	<ul style="list-style-type: none"> ● History of Computer Science ● Computer Science as a field of study ● Ethical use of computers ● Social implications of computers 	Classroom Activities Unit Test	ISTE 5.a ISTE 6.a
	2	Students will understand how hardware and software components of a computer system interrelate and function.	L	<ul style="list-style-type: none"> ● Hardware components and binary numbers ● Software components and language development ● Software engineering concepts 	Classroom Activities Unit Test	ISTE 5.a ISTE 6.a
	3	Students will run an existing C++ program in order to understand the editing, compiling, and linking process and be able to read and understand the actual code.	U	<ul style="list-style-type: none"> ● Editing, Compiling, and Linking ● Debugging and Testing ● Understand structured code 	Programming project - Factoring Unit Test	ISTE 6.a ISTE 4.b
II. Control Structures in C++	4	Students will know how to write a simple main method with including libraries, declaring variables, and using console input and output.	U	<ul style="list-style-type: none"> ● Using cout to output information ● Using cin to input information ● Formatting output with setw and setprecision from iomanip.h ● All kinds of assignment statements 	Class Presentations Programming project - The Chaos game - The Mandelbrot Set Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	5	Students will be able to do mixed-mode calculations using Boolean, relational, and arithmetic operators or functions along with C++ shortcuts.		<ul style="list-style-type: none"> ● Arithmetic (+, -, *, /, %), relational (>, <, ==, !=, >=, <=), and Boolean operators (&&, , !) ● Shortcuts (++, --, +=, -=, *=, /=, %=) ● Functions (pow, abs, sqrt) ● Type-casting between data types 	Class Presentations Programming project - The Chaos game - The Mandelbrot Set Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	6	Students will know how to generate random numbers in a C++ program.		<ul style="list-style-type: none"> ● Creating a random seed with srand by using the system clock. ● Getting a sequence of random numbers within a specified range with the rand function 	Class Presentations Programming project - The Chaos game - The Mandelbrot Set Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	7	Students will know how to write a program with three kinds of loop structures for the purpose of repeating a section of code.	U	<ul style="list-style-type: none"> ● Definite loops with the for statement ● Indefinite loops with while or do...while statements ● Nested loop structures 	Class Presentations Programming project - The Chaos game - The Mandelbrot Set Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d

	8	Students will know how to write a program with two kinds of decision structures for the purpose of deciding among two or more alternatives.	U	<ul style="list-style-type: none"> ● One-way, two-way, and multi-way decisions with the if...then...else statement ● Specific matching with the switch...case statement 	Class Presentations Programming project <ul style="list-style-type: none"> - The Chaos game - The Mandelbrot Set Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
III. Strings, Files, and Arrays	9	Students will understand the use of classes for object-oriented programming including apstring , apvector , and apmatrix .	U	<ul style="list-style-type: none"> ● Understanding how to use the apstring, apvector, and apmatrix header files without caring about their implementation ● Understanding templated classes 	Class Presentations Programming project <ul style="list-style-type: none"> - The Hangman game - The Game of Reverse Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	10	Students will know how to find individual characters in a string and solve related problems.	U	<ul style="list-style-type: none"> ● Using the length and substr methods ● Using the [] operator to get individual characters ● String processing algorithms 	Class Presentations Programming project <ul style="list-style-type: none"> - The Hangman game - The Game of Reverse Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	11	Students will know how to input from and output to an external file.	U	<ul style="list-style-type: none"> ● Declaring a file and using the open and close methods ● Inputting using getline or the >> operator ● Outputting using the << operator 	Class Presentations Programming project <ul style="list-style-type: none"> - The Hangman game - The Game of Reverse Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	12	Students will input, output, and process one-dimensional arrays while understanding how they are stored internally.	U	<ul style="list-style-type: none"> ● When arrays are needed for solving problems ● Inputting arrays using a counter or sentinel loop ● Outputting arrays with formatting 	Class Presentations Programming project <ul style="list-style-type: none"> - The Hangman game - The Game of Reverse Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	13	Students will solve problems involving arrays such as table look-up, frequency distribution, searching, sorting, reversing, etc.	U	<ul style="list-style-type: none"> ● Frequency distribution problems ● Table look-up problems ● Searching and sorting problems ● Reversing and merging problems 	Class Presentations Programming project <ul style="list-style-type: none"> - The Hangman game - The Game of Reverse Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
IV. Two-dimensional Arrays and User-defined Functions	14	Students will input, output, and process multi-dimensional arrays while understanding how they are stored internally.	R	<ul style="list-style-type: none"> ● Inputting and outputting 2D arrays ● Accumulating data in 2D arrays ● Traversing and modifying 2D arrays 	Class Presentations Programming project <ul style="list-style-type: none"> - The Game of Tic Tac Toe - The Game of Othello Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d

	15	Students will use their own functions to modularize their code by writing functions of their own design.	U	<ul style="list-style-type: none"> ● Methods that perform a task (procedures) versus return a value (functions) ● Calling methods from the main method 	Class Presentations Programming project <ul style="list-style-type: none"> - The Game of Tic Tac Toe - The Game of Othello Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	16	Students will use call-by-value and call-by-reference parameters to pass values or reference variable locations in the main function.	U	<ul style="list-style-type: none"> ● Call-by-value versus call-by-reference parameters ● Overloaded functions and operators 	Class Presentations Programming project <ul style="list-style-type: none"> - The Game of Tic Tac Toe - The Game of Othello Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
V. Building Structured Data with Structs, Classes, and Objects	17	Students will design their own structs and redefine them as classes with attributes of and behaviors for its objects.	U	<ul style="list-style-type: none"> ● Understanding attributes and behaviors ● Differences between structs and objects of classes 	Class Presentations Programming project <ul style="list-style-type: none"> - Turtle and Hare Race - Monte Carlo Calculations Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	18	Students will know how to explain the concepts of object-oriented design, encapsulation, information hiding, polymorphism, and inheritance.	U	<ul style="list-style-type: none"> ● Object-oriented design benefits ● Encapsulation of private attributes and public methods ● Inheriting attributes and behaviors from another class 	Class Presentations Programming project <ul style="list-style-type: none"> - Turtle and Hare Race - Monte Carlo Calculations Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	19	Students will use already-created classes in a client program.	U	<ul style="list-style-type: none"> ● Constructing objects of the class ● Using methods of the class ● Using operators of the class ● Destructing objects of the class 	Class Presentations Programming project <ul style="list-style-type: none"> - Turtle and Hare Race - Monte Carlo Calculations Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	20	Students will create classes of their own design including constructors, destructors, accessors, and mutators.	U	<ul style="list-style-type: none"> ● Defining constructors and destructors ● Defining methods that might be overloaded ● Defining overloaded operators 	Class Presentations Programming project <ul style="list-style-type: none"> - Turtle and Hare Race - Monte Carlo Calculations Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
VI. Advanced Data Structures with Reference Pointers	21	Students will understand the nature of reference pointers and a linked list and binary tree abstract data structure.	U	<ul style="list-style-type: none"> ● Declaring and using reference pointers as variables ● Linked lists concepts ● Binary search tree concepts 	Class Presentations Programming project <ul style="list-style-type: none"> - The Game of 20 Questions - Multiple Choice Tutorial Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d

	22	Students will create a LinkedList class that will use objects of a ListNode class	U	<ul style="list-style-type: none"> ● Adding, deleting, searching, and printing nodes in a linked list 	Class Presentations Programming project <ul style="list-style-type: none"> - The Game of 20 Questions - Multiple Choice Tutorial Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	23	Students will create a BinaryTree class that will use objects of a TreeNode class.	U	<ul style="list-style-type: none"> ● Binary tree terminology ● Adding, deleting, searching, and printing nodes in a binary search tree ● 	Class Presentations Programming project <ul style="list-style-type: none"> - The Game of 20 Questions - Multiple Choice Tutorial Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d
	24	Students will solve recursive problems using Linked Lists and Binary Trees.		<ul style="list-style-type: none"> ● Recursive processing ● Algorithm design with linked lists ● Algorithm design with binary trees ● Efficiency of recursive algorithms 	Class Presentations Programming project <ul style="list-style-type: none"> - The Game of 20 Questions - Multiple Choice Tutorial Unit Test	ISTE 1.b ISTE 3.b ISTE 3.d ISTE 4.b ISTE 6.d