Computer Science - C++

Length of Course: Term
Elective / Required: Elective
Schools: High Schools
Student Eligibility: Grade 10-12
Credit Value: 5 Credits
Date Approved: 6/14/10
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement of Purpose</td>
<td>3</td>
</tr>
<tr>
<td>Introduction</td>
<td>4</td>
</tr>
<tr>
<td>Course Objectives</td>
<td>6</td>
</tr>
<tr>
<td>Suggested Time Schedule</td>
<td>7</td>
</tr>
<tr>
<td>Unit 1: Introduction to Computers</td>
<td>8</td>
</tr>
<tr>
<td>Unit 2: How Computers are Programmed</td>
<td>9</td>
</tr>
<tr>
<td>Unit 3: Entering, Compiling and Running a Program</td>
<td>10</td>
</tr>
<tr>
<td>Unit 4: Variables and Constants</td>
<td>11</td>
</tr>
<tr>
<td>Unit 5: Math Operations</td>
<td>12</td>
</tr>
<tr>
<td>Unit 6: Strings and Screen I/O</td>
<td>13</td>
</tr>
<tr>
<td>Unit 7: Decision Making in Programs</td>
<td>14</td>
</tr>
<tr>
<td>Unit 7: Loops</td>
<td>15</td>
</tr>
<tr>
<td>Unit 8: Functions</td>
<td>16</td>
</tr>
<tr>
<td>Unit 9: Arrays and the Vector Class</td>
<td>17</td>
</tr>
<tr>
<td>Unit 10: Multi-Dimensional Arrays and Matrices</td>
<td>18</td>
</tr>
<tr>
<td>Course Requirements</td>
<td>19</td>
</tr>
<tr>
<td>Edison’s Framework for Essential Instructional Behaviors (Draft 14)</td>
<td>21</td>
</tr>
</tbody>
</table>
This course is designed to extend the student’s understanding of programming. With the increased use of object-oriented languages as a means to solve real world problems in the fields of mathematics, science and business, there exists a need for the student to understand the capabilities and limitations of these languages. The expected outcome of Computer Science - C++ is to introduce the use of object-oriented programming to the student, using the C++ language as a model.

This guide was revised/updated by:

Florene Quan - John P. Stevens High School

Coordinated by:

Donald Jobbins, Supervisor - Edison High School
Vincent Ciraulo, Supervisor - John P. Stevens High School
Introduction

The most precious resource teachers have is time. Regardless of how much time a course is scheduled for, it is never enough to accomplish all that one would like. Therefore, it is imperative that teachers utilize the time they have wisely in order to maximize the potential for all students to achieve the desired learning.

High quality educational programs are characterized by clearly stated goals for student learning, teachers who are well-informed and skilled in enabling students to reach those goals, program designs that allow for continuous growth over the span of years of instruction, and ways of measuring whether students are achieving program goals.

The Edison Township School District Curriculum Template

The Edison Township School District has embraced the backward-design model as the foundation for all curriculum development for the educational program. When reviewing curriculum documents and the Edison Township curriculum template, aspects of the backward-design model will be found in the stated enduring understandings/essential questions, unit assessments, and instructional activities. Familiarization with backward-design is critical to working effectively with Edison’s curriculum guides.

Guiding Principles: What is Backward Design? What is Understanding by Design?

“Backward design” is an increasingly common approach to planning curriculum and instruction. As its name implies, “backward design” is based on defining clear goals, providing acceptable evidence of having achieved those goals, and then working “backward” to identify what actions need to be taken that will ensure that the gap between the current status and the desired status is closed.

Building on the concept of backward design, Grant Wiggins and Jay McTighe (2005) have developed a structured approach to planning programs, curriculum, and instructional units. Their model asks educators to state goals; identify deep understandings, pose essential questions, and specify clear evidence that goals, understandings, and core learning have been achieved.

Program based on backward design use desired results to drive decisions. With this design, there are questions to consider, such as: What should students understand, know, and be able to do? What does it look like to meet those goals? What kind of program will result in the outcomes stated? How will we know students have achieved that result? What other kinds of evidence will tell us that we have a quality program? These questions apply regardless of whether they are goals in program planning or classroom instruction.
The backward design process involves three interrelated stages for developing an entire curriculum or a single unit of instruction. The relationship from planning to curriculum design, development, and implementation hinges upon the integration of the following three stages.

**Stage I: Identifying Desired Results:** Enduring understandings, essential questions, knowledge and skills need to be woven into curriculum publications, documents, standards, and scope and sequence materials. Enduring understandings identify the “big ideas” that students will grapple with during the course of the unit. Essential questions provide a unifying focus for the unit and students should be able to answer more deeply and fully these questions as they proceed through the unit. Knowledge and skills are the “stuff” upon which the understandings are built.

**Stage II: Determining Acceptable Evidence:** Varied types of evidence are specified to ensure that students demonstrate attainment of desired results. While discrete knowledge assessments (e.g.: multiple choice, fill-in-the-blank, short answer, etc…) will be utilized during an instructional unit, the overall unit assessment is performance-based and asks students to demonstrate that they have mastered the desired understandings. These culminating (summative) assessments are authentic tasks that students would likely encounter in the real-world after they leave school. They allow students to demonstrate all that they have learned and can do. To demonstrate their understandings students can explain, interpret, apply, provide critical and insightful points of view, show empathy and/or evidence self-knowledge. Models of student performance and clearly defined criteria (i.e.: rubrics) are provided to all students in advance of starting work on the unit task.

**Stage III: Designing Learning Activities:** Instructional tasks, activities, and experiences are aligned with stages one and two so that the desired results are obtained based on the identified evidence or assessment tasks. Instructional activities and strategies are considered only once stages one and two have been clearly explicated. Therefore, congruence among all three stages can be ensured and teachers can make wise instructional choices.

At the curricular level, these three stages are best realized as a fusion of research, best practices, shared and sustained inquiry, consensus building, and initiative that involves all stakeholders. In this design, administrators are instructional leaders who enable the alignment between the curriculum and other key initiatives in their district or schools. These leaders demonstrate a clear purpose and direction for the curriculum within their school or district by providing support for implementation, opportunities for revision through sustained and consistent professional development, initiating action research activities, and collecting and evaluating materials to ensure alignment with the desired results. Intrinsic to the success of curriculum is to show how it aligns with the overarching goals of the district, how the document relates to district, state, or national standards, what a high quality educational program looks like, and what excellent teaching and learning looks like. Within education, success of the educational program is realized through this blend of commitment and organizational direction.
COURSE OBJECTIVES

At the successful completion of this course, the student will:

1. Demonstrate an understanding of the capabilities and limitations of the microcomputer (7, 8)*

2. Demonstrate proficiency in operating a computer system (3, 4)

3. Demonstrate proficiency in advanced programming techniques using the C++ computer language (3, 4, 5, 7, 8)

4. Use the computer to solve problems which will be encountered in advanced courses in mathematics and science (3, 5, 7, 8)

*Number(s) in parentheses refer to program objectives.
## SUGGESTED TIME SCHEDULE

### SUGGESTED INSTRUCTIONAL

<table>
<thead>
<tr>
<th>UNIT</th>
<th>TIME:</th>
<th>CLASS PERIODS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction to Computers</td>
<td>5</td>
</tr>
<tr>
<td>2.</td>
<td>How Computers are Programmed</td>
<td>5</td>
</tr>
<tr>
<td>3.</td>
<td>Entering, compiling and Running a Program</td>
<td>10</td>
</tr>
<tr>
<td>4.</td>
<td>Variables and Constants</td>
<td>15</td>
</tr>
<tr>
<td>5.</td>
<td>Math Operations</td>
<td>15</td>
</tr>
<tr>
<td>6.</td>
<td>String and Screen I/O</td>
<td>20</td>
</tr>
<tr>
<td>7.</td>
<td>Decision Making</td>
<td>20</td>
</tr>
<tr>
<td>8.</td>
<td>Loops</td>
<td>25</td>
</tr>
<tr>
<td>9.</td>
<td>Functions</td>
<td>25</td>
</tr>
<tr>
<td>10.</td>
<td>Arrays and the Vector Class</td>
<td>20</td>
</tr>
<tr>
<td>11.</td>
<td>Multi-dimensional Arrays and Matrices</td>
<td>20</td>
</tr>
</tbody>
</table>

**Total Class Periods** 180
## Introduction to Computers

**Targeted Standards: Standard 8.1:** All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaboratively and to create and communicate knowledge.

**Unit Objectives/Enduring Understandings:** Students will be able to study the history of computers, how they evolved into present day desktop computers, and the basics of microcomputer architecture.

**Essential Questions:** What are major hardware and software components? How do computers operate internally?

**Unit Assessment:** Programming assignments and chapter test

### Cumulative Progress Indicators

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Skills</th>
<th>Activities/Strategies</th>
<th>Assessment Check Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>What students will know.</td>
<td>What students will be able to do.</td>
<td>Technology Implementation/Interdisciplinary Connections</td>
<td>Use section questions from textbook and student worksheet from workbook</td>
</tr>
<tr>
<td><strong>Core Content Objectives</strong></td>
<td><strong>Instructional Actions</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Create a technological product, system or environment using given design specifications and constraints by applying design and engineering principles. 8.2.B6 | • Computers are complex devices where electrical signals are used to represent data and give instructions to system components  
• The history of computers begins with calculating tools, extends to electro-mechanical devices and finally to electronic computers  
• A computer system consists of hardware and software that work together to help us solve problems  
• Secondary memory vs. main memory and its role in executing a program  
• The architecture of a computer describes the many devices that make up a computer and how they connect to solve a problem  
• A network is two or more computers connected together so they can exchange information | • List key hardware and software components  
• Report on development of computers prior to 1960  
• List pioneers in the early development of computers | • Develop PowerPoint slides to cover material in textbook  
• Select several projects from the end of the chapter |

### Resources:

- Essential Materials, Supplementary Materials, Links to Best Practices
- **Textbook and related Teacher Resources.**
- **Timeline of Computer History (’39 – ’94):** http://www.computerhistory.org/timeline/
- **A closer look at the PC:** http://www.howstuffworks.com/pc.htm

### Instructional Adjustments:

- Modifications, student difficulties, possible misunderstandings
- Circulate around classroom during Lab and offer one-on-one assistance for struggling students
Computer Science - C++  

**How are Computers Programmed**

**Targeted Standards:**  
**Standard 4.1:** All students will develop number sense and will perform standard numerical operations in a variety of ways.  
**Standard 8.2:** All students will develop an understanding of the nature and impact of technology, engineering, technological design, and the designed world, as they relate to the individual, global society, and the environment.

**Unit Objectives/Enduring Understandings:** Students will be able to understand how data and instructions are represented, the roles of assemblers, interpreters, compilers and operating systems, and be able to describe the steps in the programming process.

**Essential Questions:** How do computers receive instructions? What steps are essential to follow in order to produce good computer programs?

**Unit Assessment:** Programming assignments and chapter test

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Core Content Objectives</th>
<th>Instructional Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concepts</strong></td>
<td><strong>Skills</strong></td>
<td><strong>Activities/Strategies</strong></td>
</tr>
<tr>
<td>What students will know.</td>
<td>What students will be able to do.</td>
<td>Technology Implementation/Interdisciplinary Connections</td>
</tr>
</tbody>
</table>
| Extend understanding of the number system to all real numbers. 4.1.A1 | All data in a computer is represented in binary  
Instructions to a computer are represented in machine language, also binary  
Programming languages provide a way to program computers using instructions that can be understood by computers and humans  
There are advantages and disadvantages to using low-level vs. high-level languages  
Some programming languages use interpreters and others use compilers  
C++ is a high-level, compiled language  
The computers operating system manage the fundamental operations of the computer  
There are five basic steps when developing programs | Convert base 10 numbers into base 2 and base 16. Convert base 2 and base 16 to decimal  
Define units of storage for computer memory such as byte, kilobyte, megabyte, gigabyte and terabyte  
Categorize programming languages as low-level or high-level languages, compiled or interpreted.  
Identify functions of the operating system  
List the five basic steps to developing programs | Develop PowerPoint slides to cover material in textbook  
Select several projects from the end of the chapter |

**Resources:**  
- Essential Materials, Supplementary Materials, Links to Best Practices  
- Textbook and related Teacher Resources.  
- A summary of programming languages - http://www.scriptol.org/history.php

**Instructional Adjustments:**  
- Modifications, student difficulties, possible misunderstandings  
- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

---

**Extend understanding of the number system to all real numbers. 4.1.A1**

- Analyze a given technological product, system, or environment to understand how the engineering design process and design specification limitations influenced the final solution. 8.2.B1
### Entering, Compiling, and Running a Program

**Targeted Standards: Standard 8.2:** All students will develop an understanding of the nature and impact of technology, engineering, technological design, and the designed world, as they relate to the individual, global society, and the environment.

**Unit Objectives/Enduring Understandings:** Students will be able to understand the structure of a C++ program and be able to enter, compile, and run a C++ program.

**Essential Questions:** How are C++ programs structured and what steps are needed to run a program?

**Unit Assessment:** Programming assignments and chapter test

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Core Content Objectives</th>
<th>Instructional Actions</th>
<th>Assessment Check Points</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concepts</strong></td>
<td><strong>Skills</strong></td>
<td><strong>Activities/Strategies</strong></td>
<td></td>
</tr>
<tr>
<td>What students will know.</td>
<td>What students will be able to do.</td>
<td>Technology Implementation/Interdisciplinary Connections</td>
<td></td>
</tr>
<tr>
<td><strong>Skills</strong></td>
<td><strong>Activities/Strategies</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>List uses for comments</td>
<td>Develop PowerPoint slides to cover material in textbook</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use the compiler directive to include source code from another file</td>
<td>Select several projects from the end of the chapter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define and describe the main function</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recognize basic syntax rules such as “case sensitive”, matching braces and the semicolon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create a C++ project in eclipse</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Textbook and related Teacher Resources.</td>
</tr>
<tr>
<td>● C++ details - <a href="http://cplus.about.com/od/introductiontoprogramming/a/cppbeginners.htm">http://cplus.about.com/od/introductiontoprogramming/a/cppbeginners.htm</a></td>
</tr>
<tr>
<td>● An online C++ tutorial - <a href="http://www.cplusplus.com/doc/tutorial/">http://www.cplusplus.com/doc/tutorial/</a></td>
</tr>
<tr>
<td>● Eclipse Software - <a href="http://www.eclipse.org/">http://www.eclipse.org/</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Circulate around classroom during Lab and offer one-on-one assistance for struggling students</td>
</tr>
</tbody>
</table>
### Variables and Constants

**Targeted Standards:** Standard 4.1: All students will develop number sense and will perform standard numerical operations in a variety of ways.

**Unit Objectives/Enduring Understandings:** Students will be able to categorize data types and recognize the difference between variable and constants.

**Essential Questions:** How can a programmer effectively organize data for storage and use?

**Unit Assessment:** Programming assignments and chapter test

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Core Content Objectives</th>
<th>Instructional Actions</th>
<th>Assessment Check Points</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concepts</strong></td>
<td><strong>Skills</strong></td>
<td><strong>Activities/Strategies</strong></td>
<td><strong>Assessment Check Points</strong></td>
</tr>
<tr>
<td>What students will know.</td>
<td>What students will be able to do.</td>
<td>Technology Implementation/ Interdisciplinary Connections</td>
<td>Use section questions from textbook and student worksheet from workbook</td>
</tr>
</tbody>
</table>
| Extend understanding of the number system to all real numbers. 4.1.A1 | ● A variable holds data that can be modified while a program is running  
● A constant stores data that remains static throughout the program’s execution  
● Appropriate use of different data types | ● Develop PowerPoint slides to cover material in textbook  
● Select several projects from the end of the chapter | |
| Extend understanding and use of operations to real numbers and algebraic procedures. 4.1.B1 | ● The following integer data types: int, short, long, unsigned int, unsigned long  
● The following floating-point data types: float, double, long double  
● Character data types: char, unsigned char  
● Boolean data type  
● Rules for declaring and naming variable | | |

**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related Teacher Resources.
- An online C++ tutorial - http://www.cplusplus.com/doc/tutorial/

**Instructional Adjustments:** Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students.
## Computer Science - C++

### Targeted Standards:
- **Standard 4.1:** All students will develop number sense and will perform standard numerical operations in a variety of ways.
- **Standard 4.3:** All students will present and analyze relationships among variable quantities and solve problems involving patterns, functions, and algebraic concepts and processes.
- **Standard 4.4:** All students will develop an understanding of the concepts and techniques of data analysis, probability, and discrete mathematics, and will use them to model situations, solve problems, and analyze and draw appropriate inferences from data.
- **Standard 4.5:** All students will use mathematical processes of problem solving, communication, connections, reasoning, representations, and technology to solve problems and communicate mathematical ideas.

### Unit Objectives/Enduring Understandings:
Students will be able to use arithmetic operators and the increment and decrement operator. Student will be able to understand order of operations and the affects of mixed data types.

### Essential Questions:
How can C++ operators be used to solve problems?

### Unit Assessment:
Programming assignments and chapter test

### Core Content Objectives

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Skills</th>
<th>Activities/Strategies</th>
<th>Assessment Check Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>What students will know.</td>
<td>What students will be able to do.</td>
<td>Technology Implementation/Interdisciplinary Connections</td>
<td>Use section questions from textbook and student worksheet from workbook</td>
</tr>
<tr>
<td>- The equal sign <code>=</code> is used to change the value of a variable</td>
<td>- Write a syntactically correct assignment statement and compound operators</td>
<td>- Develop PowerPoint slides to cover material in textbook</td>
<td>- Use section questions from textbook and student worksheet from workbook</td>
</tr>
<tr>
<td>- The percent sign <code>%</code> is used as a modulus operator</td>
<td>- Use the increment/decrement operator</td>
<td>- Select several projects from the end of the chapter</td>
<td></td>
</tr>
<tr>
<td>- Variable need to be declared and initialized</td>
<td>- Use the modulus operator</td>
<td>- Develop PowerPoint slides to cover material in textbook</td>
<td></td>
</tr>
<tr>
<td>- C++ calculations follow an order of operations</td>
<td>- Write expressions that involve type coercion and type-casting</td>
<td>- Select several projects from the end of the chapter</td>
<td></td>
</tr>
<tr>
<td>- C++ may change data types in mixed-type expressions</td>
<td></td>
<td>- Develop PowerPoint slides to cover material in textbook</td>
<td></td>
</tr>
<tr>
<td>- Dividing by zero generates an error in C++</td>
<td></td>
<td>- Select several projects from the end of the chapter</td>
<td></td>
</tr>
</tbody>
</table>

### Instructional Actions

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

### Resources:
- Essential Materials, Supplementary Materials, Links to Best Practices
- Textbook and related Teacher Resources.
## Strings and Screen I/O

**Targeted Standards: Standard 4.5:** All students will use mathematical processes of problem solving, communication, connections, reasoning, representations, and technology to solve problems and communicate mathematical ideas.

**Unit Objectives/Enduring Understandings:** Students will be able to display and retrieve data for the user and use and manipulate characters.

**Essential Questions:** How can input/output operators be used to make dynamic programs?

**Unit Assessment:** Programming assignments and chapter test

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Core Content Objectives</th>
<th>Instructional Actions</th>
<th>Assessment Check Points</th>
</tr>
</thead>
</table>
| Select and apply a variety of appropriate problem-solving strategies to solve problems. 4.5.A3 | **Concepts**
What students will know. | **Skills**
What students will be able to do. | **Activities/Strategies**
Technology Implementation/Interdisciplinary Connections | **Assessment Check Points**
Use section questions from textbook and student worksheet from workbook |
| Apply mathematics in practical situations and in other disciplines. 4.5.C4 | | | |
| Select, apply and translate among mathematical representations to solve problems. 4.5.E2 | | | |
| | ● There is a distinction between a string literal and an array of characters  
● Recognize misuse of the assignment statement with character data  
● Using input/output commands  
● Displaying data in visually pleasing format | ● Display a string literal  
● Declare an array of characters  
● Correct use of the null terminator “\0”  
● Use the strcpy function  
● Using cin >> and cout <<  
● Using escape sequences to tab “\t”, to skip a line “\n” and to print a single quote and a double quote  
● Use iomanip.h to format output | ● Develop PowerPoint slides to cover material in textbook  
● Select several projects from the end of the chapter | |

**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related Teacher Resources.
- An online C++ tutorial - http://www.cplusplus.com/doc/tutorial/

**Instructional Adjustments:** Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students
### Decision Making in Programs

**Targeted Standards: Standard 4.5:** All students will use mathematical processes of problem solving, communication, connections, reasoning, representations, and technology to solve problems and communicate mathematical ideas.

**Unit Objectives/Enduring Understandings:** Students will be able to use control statements to manipulate execution of a program.

**Essential Questions:** How can decision-making be used to enhance execution of a C++ program?

**Unit Assessment:** Programming assignments and chapter test

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Concepts</th>
<th>Skills</th>
<th>Activities/Strategies</th>
<th>Assessment Check Points</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Learn mathematics through problem solving, inquiry, and discovery.</strong> 4.5.A1</td>
<td>Control structures modify the flow of control in any programming language</td>
<td>Use the set of relational operators including &lt;, &lt;=, &gt;=, ==, !=</td>
<td>Develop PowerPoint slides to cover material in textbook</td>
<td>Use section questions from textbook and student worksheet from workbook</td>
</tr>
<tr>
<td><strong>Select and apply a variety of problem-solving strategies to solve problems.</strong> 4.5.A3</td>
<td>Relational operators are used to compare</td>
<td>Use the three logical operators including &amp;&amp;,</td>
<td></td>
<td>, and !</td>
</tr>
<tr>
<td><strong>Apply mathematics in practical situation and in other disciplines.</strong> 4.5.C4</td>
<td>False is represented by integer 0 and True is represented by integer 1</td>
<td>Use the if-then and if-then-else statements</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Logical operators combine multiple boolean expressions</td>
<td>Use nested if structures</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C++ selection statements use short-circuit evaluation</td>
<td>Use the switch statement</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Core Content Objectives**
- Use the set of relational operators including <, <=, >=, ==, !=
- Use the three logical operators including &&, ||, and !
- Use the if-then and if-then-else statements
- Use nested if structures
- Use the switch statement

**Instructional Adjustments:** Modifications, student difficulties, possible misunderstandings
- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices
- Textbook and related Teacher Resources.
- An online C++ tutorial - http://www.cplusplus.com/doc/tutorial/
## Loops

### Targeted Standards: Standard 4.3
All students will present and analyze relationships among variable quantities and solve problems involving patterns, functions, and algebraic concepts and processes.

### Unit Objectives/Enduring Understandings
Students will be able to use three types of iteration structures.

### Essential Questions
How can the computer's effective use of iteration be used to solve problems?

### Unit Assessment
Programming assignments and chapter test

### Core Content Objectives

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Concepts What students will know</th>
<th>Skills What students will be able to do</th>
<th>Activities/Strategies Technology Implementation/Interdisciplinary Connections</th>
<th>Assessment Check Points</th>
</tr>
</thead>
</table>
| Use functions to model real-world phenomena and solve problems that involve varying quantities. 4.3.C1 | ● There are three iteration structures in C++  
● Some loops are counter controlled and others are event controlled  
● An infinite loop will cause unpredictable results | ● Use the for loop, the while loop and the do-while loop  
● Use the break and continue statement  
● Use nested loops | ● Develop PowerPoint slides to cover material in textbook  
● Select several projects from the end of the chapter | ● Use section questions from textbook and student worksheet from workbook |
| Analyze and describe how a change in an independent variable leads to change in a dependent one. 4.3.C2 |  |  |  |  |
| Convert recursive formulas to linear or exponential functions. 4.3.C3 |  |  |  |  |

### Resources: Essential Materials, Supplementary Materials, Links to Best Practices
- Textbook and related Teacher Resources.

### Instructional Adjustments: Modifications, student difficulties, possible misunderstandings
- Circulate around classroom during Lab and offer one-on-one assistance for struggling students
## Targeted Standards: Standard 4.3

All students will present and analyze relationships among variable quantities and solve problems involving patterns, functions, and algebraic concepts and processes.

## Unit Objectives/Enduring Understandings

Students will be able to modularize programs using functions.

## Essential Questions

What are the benefits of sub-dividing programs into more manageable tasks?

## Unit Assessment

Programming assignments and chapter test

### Core Content Objectives

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>What students will know.</td>
<td>What students will be able to do.</td>
</tr>
</tbody>
</table>

### Activities/Strategies

Technology Implementation/Interdisciplinary Connections

- Develop PowerPoint slides to cover material in textbook
- Select several projects from the end of the chapter

### Assessment Check Points

- Use section questions from textbook and student worksheet from workbook

### Instructional Adjustments

- Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

### Cumulative Progress Indicators

- Use functions to model real-world phenomena and solve problems that involve varying quantities. 4.3.C1
  - Structured programs are built using functions
  - A Visual Table Of Content will display the hierarchy of functions
  - A well-organized program is organized into functions
  - Encapsulation refers to enclosing the details of a function within the function itself.
  - Functions from one application can be reused in other applications
  - Variable in a program have a scope and lifetime
  - Data can be passed back and forth between the calling program and the function

- Call pre-written library functions
- Write a function including the function prototype, the function heading and the function body and the invoking statement
- Use pass-by-value and pass-by-reference parameters
- Use the return statement

### Resources

- Essential Materials, Supplementary Materials, Links to Best Practices
  - Textbook and related Teacher Resources.
  - An online C++ tutorial - http://www.cplusplus.com/doc/tutorial/

## Functions

- Use written functions to model real-world phenomena and solve problems involving varying quantities. 4.3.C1
- Structured programs are built using functions
- A Visual Table Of Content will display the hierarchy of functions
- A well-organized program is organized into functions
- Encapsulation refers to enclosing the details of a function within the function itself.
- Functions from one application can be reused in other applications
- Variable in a program have a scope and lifetime
- Data can be passed back and forth between the calling program and the function

- Call pre-written library functions
- Write a function including the function prototype, the function heading and the function body and the invoking statement
- Use pass-by-value and pass-by-reference parameters
- Use the return statement
### Arrays and the Vector class

**Targeted Standards: Standard 4.4:** All students will develop an understanding of the concepts and techniques of data analysis, probability, and discrete mathematics, and will use them to model situations, solve problems, and analyze and draw appropriate inferences from data.

**Unit Objectives/Enduring Understandings:** Students will be able use arrays and the vector class to store and manipulate large amounts of data.

**Essential Questions:** How can C++ be used to store, and manipulate large amounts of data?

**Unit Assessment:** Programming assignments and chapter test

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Concepts</th>
<th>Skills</th>
<th>Activities/Strategies</th>
<th>Assessment Check Points</th>
</tr>
</thead>
</table>
| Evaluate the use of data in real-world contexts. 4.4.A2 | ● A one-dimensional array can store large amounts of related data under one name  
● Each data item in an array is called an element  
● The position of each element is maintained using the subscript or index  
● A standard C++ array requires the programmer to avoid data to be written beyond the boundaries of the array  
● Use of the Vector class can prevent accessing data outside of the array | ● Declare and initialize a one-dimensional array  
● Use a loop to populate and traverse through an array  
● Create and populate an array using an initializer list  
● Create an array using the vector class | ● Develop PowerPoint slides to cover material in textbook  
● Select several projects from the end of the chapter | ● Use section questions from textbook and student worksheet from workbook |

**Core Content Objectives**

**Concepts**  
What students will know.

**Skills**  
What students will be able to do.

**Activities/Strategies**  
Technology Implementation/Interdisciplinary Connections

**Assessment Check Points**

**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related Teacher Resources.

**Instructional Adjustments:** Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

---

**Notes:**
**Multi-Dimensional Arrays and Matrices**

**Targeted Standards: Standard 4.4:** All students will develop an understanding of the concepts and techniques of data analysis, probability, and discrete mathematics, and will use them to model situations, solve problems, and analyze and draw appropriate inferences from data.

**Unit Objectives/Enduring Understandings:** Students will be able use arrays and the matrix class to store and manipulate large amounts of data.

**Essential Questions:** How can C++ be used to store, and manipulate large amounts of data?

**Unit Assessment:** Programming assignments and chapter test

<table>
<thead>
<tr>
<th>Cumulative Progress Indicators</th>
<th>Concepts</th>
<th>Skills</th>
<th>Activities/Strategies</th>
<th>Assessment Check Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluate the use of data in real-world contexts. 4.4.A2</td>
<td>Tabular data can be stored in a chart-form or a matrix using parallel arrays or a multi-dimensional array. Tabular data is stored in rows and columns. C++ reports the size of arrays to the programmer. Use of the matrix class to prevent access beyond the boundaries of the array.</td>
<td>Declare and populate parallel arrays and two-dimensional arrays. Discuss applications for multi-dimensional arrays. Use the size of operator to prevent access outside the boundaries of the array.</td>
<td>Develop PowerPoint slides to cover material in textbook. Select several projects from the end of the chapter.</td>
<td>Use section questions from textbook and student worksheet from workbook.</td>
</tr>
</tbody>
</table>

**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related Teacher Resources.
- An online C++ tutorial - http://www.cplusplus.com/doc/tutorial/

**Instructional Adjustments:** Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students.
I. **COURSE CONTENT** - This course will consist of the following units of study:

A. Operating procedures for C++ system protocol, use of an IDE, printer control.

B. Fundamentals of the C++ Language: C++ character set, identifiers, assignment statements, basic operations, order of operation, int, unsigned int, short int, unsigned short int, long int, unsigned long int, character, signed and unsigned character, float, double, long double, void, boolean, # include, write programs.

C. C++ Input and Output: input stream, output stream, local variables in functions, static variables in functions, precision, width, applications.

D. Conditional Statements in C++: Boolean expressions, IF-THEN statements, IF-THEN-ELSE statements, nested IF statements, switch statements, short-circuit evaluation, program writing.

E. Loop Control in C++: FOR statements, DO..WHILE statements, WHILE statements, continue statements, skipping loop iterations, exiting loops, nested loops, flags, accumulators, formatted output, test data, applications.

F. C++ Functions and Parameters: Function prototypes, headings, parameters, interface and function invocation, program writing

G. Simple Data Types – Built-in and User-Defined: internal and external representation, numeric precision, enum, typedef, type casting, program writing

H. C++ Arrays – declaring, initializing, traversing, manipulating, calculating, printing from one and two dimensional arrays, program writing

II. **COURSE REQUIREMENTS** - To complete this course successfully, students will be required to demonstrate a satisfactory (or higher) level of proficiency in:

A. Identifying the capabilities and limitations of the computer.

B. Operating a computer system.

C. Writing introductory programs using the C++ languages.

D. Using the computer to solve problems that will be encountered in advanced courses in mathematics and science.
iii. EVALUATION PROCESS - Throughout the length of this course, students will be evaluated on the basis of:

A. Performance assessment by programming
B. Independent projects
C. Homework assignments
D. Class participation
E. Proper interface with computer

Additionally, students will maintain a folder and/or notebook, which will contain notes, projects and homework assignments. These will be reviewed periodically.
Essential Instructional Behaviors

Edison's Essential Instructional Behaviors are a collaboratively developed statement of effective teaching from pre-school through Grade 12. This statement of instructional expectations is intended as a framework and overall guide for teachers, supervisors, and administrators; its use as an observation checklist is inappropriate.

1. Planning which Sets the Stage for Learning and Assessment

**Does the planning show evidence of:**

a. units and lessons directly related to learner needs, the written curriculum, the New Jersey Core Content Curriculum Standards (NJCCCS), and the Cumulative Progress Indicators (CPI)?

b. measurable objectives that are based on diagnosis of learner needs and readiness levels and reflective of the written curriculum, the NJCCCS, and the CPI?

b. if lesson design sequenced to make meaningful connections to overarching concepts and essential questions?

d. provision for effective use of available materials, technology and outside resources?

e. accurate knowledge of subject matter?

f. multiple means of formative and summative assessment, including performance assessment, that are authentic in nature and realistically measure learner understanding?

g. differentiation of instructional content, processes and/or products reflecting differences in learner interests, readiness levels, and learning styles?

h. provision for classroom furniture and physical resources to be arranged in a way that supports student interaction, lesson objectives, and learning activities?

2. Observed Learner Behavior that Leads to Student Achievement

**Does the lesson show evidence of:**

a. learners actively engaged throughout the lesson in on-task learning activities?

b. learners engaged in authentic learning activities that support reading such as read alouds, guided reading, and independent reading utilizing active reading strategies to deepen comprehension (for example inferencing, predicting, analyzing, and critiquing)?

c. learners engaged in authentic learning activities that promote writing such as journals, learning logs, creative pieces, letters, charts, notes, graphic organizers and research reports that connect to and extend learning in the content area?

d. learners engaged in authentic learning activities that promote listening, speaking, viewing skills and strategies to understand and interpret audio and visual media?

e. learners engaged in a variety of grouping strategies including individual conferences with the teacher, learning partners, cooperative learning structures, and whole-class discussion?

f. learners actively processing the lesson content through closure activities throughout the lesson?

g. learners connecting lesson content to their prior knowledge, interests, and personal lives?

h. learners demonstrating increasingly complex levels of understanding as evidenced through their growing perspective, empathy, and self-knowledge as they relate to the academic content?

i. learners developing their own voice and increasing independence and responsibility for their learning?

j. learners receiving appropriate modifications and accommodations to support their learning?
3. Reflective Teaching which Informs Instruction and Lesson Design

Does the instruction show evidence of:

a. differentiation to meet the needs of all learners, including those with Individualized Education Plans?
b. modification of content, strategies, materials and assessment based on the interest and immediate needs of students during the lesson?
c. formative assessment of the learning before, during, and after the lesson, to provide timely feedback to learners and adjust instruction accordingly?
d. the use of formative assessment by both teacher and student to make decisions about what actions to take to promote further learning?
e. use of strategies for concept building including inductive learning, discovery-learning and inquiry activities?
f. use of prior knowledge to build background information through such strategies as anticipatory set, K-W-L, and prediction brainstorms?
g. deliberate teacher modeling of effective thinking and learning strategies during the lesson?
h. understanding of current research on how the brain takes in and processes information and how that information can be used to enhance instruction?
i. awareness of the preferred informational processing strategies of learners who are technologically sophisticated and the use of appropriate strategies to engage them and assist their learning?
j. activities that address the visual, auditory, and kinesthetic learning modalities of learners?
k. use of questioning strategies that promote discussion, problem solving, and higher levels of thinking?
l. use of graphic organizers and hands-on manipulatives?
m. creation of an environment which is learner-centered, content rich, and reflective of learner efforts in which children feel free to take risks and learn by trial and error?
n. development of a climate of mutual respect in the classroom, one that is considerate of and addresses differences in culture, race, gender, and readiness levels?
o. transmission of proactive rules and routines which students have internalized and effective use of relationship-preserving desists when students break rules or fail to follow procedures?

4. Responsibilities and Characteristics which Help Define the Profession

Does the teacher show evidence of:

a. continuing the pursuit of knowledge of subject matter and current research on effective practices in teaching and learning, particularly as they tie into changes in culture and technology?
b. maintaining accurate records and completing forms/reports in a timely manner?
c. communicating with parents about their child’s progress and the instructional process?
d. treating learners with care, fairness, and respect?
e. working collaboratively and cooperatively with colleagues and other school personnel?
f. presenting a professional demeanor?