

# Programming Fundamentals - A Modular Structured Approach using C++

**By:**

Kenneth Leroy Busbee



# Programming Fundamentals - A Modular Structured Approach using C++

**By:**

Kenneth Leroy Busbee

**Online:**

< <http://cnx.org/content/col10621/1.22/> >

**C O N N E X I O N S**

Rice University, Houston, Texas

This selection and arrangement of content as a collection is copyrighted by Kenneth Leroy Busbee. It is licensed under the Creative Commons Attribution 3.0 license (<http://creativecommons.org/licenses/by/3.0/>).

Collection structure revised: January 10, 2013

PDF generated: January 10, 2013

For copyright and attribution information for the modules contained in this collection, see p. 318.

# Table of Contents

<b>Preface</b> .....	1
<b>Author Acknowledgements</b> .....	5
<b>Orientation and Syllabus</b> .....	7
<b>Sharing/Rating Connexions Materials</b> .....	11
<b>1 1. Introduction to Programming</b>	
1.1 Systems Development Life Cycle .....	13
1.2 Bloodshed Dev-C++ 5 Compiler/IDE .....	15
1.3 Modularization and C++ Program Layout .....	18
1.4 Practice 1: Introduction to Programming .....	23
Solutions .....	26
<b>2 2. Program Planning &amp; Design</b>	
2.1 Program Design .....	27
2.2 Pseudocode .....	28
2.3 Test Data .....	29
2.4 Practice 2: Program Planning & Design .....	32
Solutions .....	35
<b>3 3. Data &amp; Operators</b>	
3.1 Data Types in C++ .....	37
3.2 Identifier Names .....	39
3.3 Constants and Variables .....	40
3.4 Data Manipulation .....	41
3.5 Assignment Operator .....	42
3.6 Arithmetic Operators .....	43
3.7 Data Type Conversions .....	44
3.8 Practice 3: Data & Operators .....	47
Solutions .....	50
<b>4 4. Often Used Data Types</b>	
4.1 Integer Data Type .....	51
4.2 Floating-Point Data Type .....	52
4.3 String Data Type .....	53
4.4 Arithmetic Assignment Operators .....	54
4.5 Lvalue and Rvalue .....	55
4.6 Integer Division and Modulus .....	57
4.7 Practice 4: Often Used Data Types .....	60
Solutions .....	62
<b>5 5. Integrated Development Environment</b>	
5.1 Integrated Development Environment .....	63
5.2 Standard Input and Output .....	69
5.3 Compiler Directives .....	72
5.4 Practice 5: Integrated Development Environment .....	74
Solutions .....	77
<b>6 6. Program Control Functions</b>	
6.1 Pseudocode Examples for Functions .....	79
6.2 Hierarchy or Structure Chart .....	80
6.3 Program Control Functions .....	81

6.4	Void Data Type	83
6.5	Documentation and Making Source Code Readable	84
6.6	Practice 6: Program Control Functions	89
	Solutions	94
<b>7</b>	<b>7. Specific Task Functions</b>	
7.1	Specific Task Functions	95
7.2	Global vs Local Data Storage	96
7.3	Using a Header File for User Defined Specific Task Functions	97
7.4	Practice 7: Specific Task Functions	103
	Solutions	105
<b>8</b>	<b>8. Standard Libraries</b>	
8.1	Standard Libraries	107
8.2	Practice 8: Standard Libraries	109
	Solutions	111
<b>9</b>	<b>9. Character Data, Sizeof, Typedef, Sequence</b>	
9.1	Character Data Type	113
9.2	Sizeof Operator	114
9.3	Typedef - An Alias	115
9.4	Sequence Operator	116
9.5	Practice 9: Character Data, Sizeof, Typedef, Sequence	118
	Solutions	121
<b>10</b>	<b>10. Introduction to Structured Programming</b>	
10.1	Structured Programming	123
10.2	Pseudocode Examples for Control Structures	125
10.3	Flowcharting	127
10.4	Practice 10: Introduction to Structured Programming	142
	Solutions	144
<b>11</b>	<b>11. Two Way Selection</b>	
11.1	If Then Else	145
11.2	Boolean Data Type	147
11.3	Relational Operators	148
11.4	Compound Statement	149
11.5	Practice 11: Two Way Selection	151
	Solutions	153
<b>12</b>	<b>12. Multiway Selection</b>	
12.1	Nested If Then Else	155
12.2	Logical Operators	158
12.3	Case Control Structure	161
12.4	Branching Control Structures	165
12.5	Practice 12: Multiway Selection	169
	Solutions	172
<b>13</b>	<b>13. Test After Loops</b>	
13.1	Do While Loop	173
13.2	Flag Concept	176
13.3	Assignment vs Equality within C++	178
13.4	Repeat Until Loop	179
13.5	Practice 13: Test After Loops	181
	Solutions	183

<b>14</b>	<b>14. Test Before Loops</b>	
14.1	Increment and Decrement Operators	185
14.2	While Loop	187
14.3	Practice 14: Test Before Loops	193
	Solutions	195
<b>15</b>	<b>15. Counting Loops</b>	
15.1	For Loop	197
15.2	Circular Nature of the Integer Data Type Family	201
15.3	Formatting Output	203
15.4	Nested For Loops	205
15.5	Practice 15: Counting Loops	210
	Solutions	212
<b>16</b>	<b>16. String Class, Unary Positive and Negative</b>	
16.1	String Class within C++	213
16.2	Unary Positive and Negative Operators	215
16.3	Practice 16: String Class, Unary Positive and Negative	217
	Solutions	219
<b>17</b>	<b>17. Conditional Operator and Recursion</b>	
17.1	Conditional Operator	221
17.2	Recursion vs Iteration	222
17.3	Practice 17: Conditional Operator and Recursion	224
	Solutions	226
<b>18</b>	<b>18. Introduction to Arrays</b>	
18.1	Array Data Type	227
18.2	Array Index Operator	228
18.3	Displaying Array Members	229
18.4	Practice 18: Introduction to Arrays	232
	Solutions	234
<b>19</b>	<b>19. File I/O and Array Functions</b>	
19.1	File Input and Output	235
19.2	Arrays and Functions	237
19.3	Loading an Array from a File	238
19.4	Math Statistics with Arrays	239
19.5	Practice 19: File I/O and Array Functions	241
	Solutions	243
<b>20</b>	<b>20. More Array Functions</b>	
20.1	Finding a Specific Member of an Array	245
20.2	Sorting an Array	246
20.3	Practice 20: More Array Functions	249
	Solutions	251
<b>21</b>	<b>21. More on Typedef</b>	
21.1	Versatile Code with Typedef	253
21.2	Practice 21: More on Typedef	255
	Solutions	257
<b>22</b>	<b>22. Pointers</b>	
22.1	Address Operator	259
22.2	Parameter Passing by Reference	260

<b>22.3</b>	Pointer Data Type .....	261
<b>22.4</b>	Indirection Operator .....	262
<b>22.5</b>	Practice 22: Pointers .....	264
	Solutions .....	266
<b>23</b>	<b>23. More Arrays &amp; Compiler Directives</b>	
<b>23.1</b>	Multidimensional Arrays .....	267
<b>23.2</b>	Conditional Compilation .....	268
<b>23.3</b>	Practice 23: More Arrays & Compiler Directives .....	270
	Solutions .....	272
<b>24</b>	<b>24. OOP &amp; HPC</b>	
<b>24.1</b>	Object Oriented Programming .....	273
<b>24.2</b>	Understanding High Performance Computing .....	274
<b>24.3</b>	Practice 24: OOP & HPC .....	281
	Solutions .....	283
<b>25</b>	<b>Review Materials</b>	
<b>25.1</b>	Review: Foundation Topics Group: 1-5 .....	286
<b>25.2</b>	Review: Modular Programming Group: 6-9 .....	287
<b>25.3</b>	Review: Structured Programming Group: 10-16 .....	288
<b>25.4</b>	Review: Intermediate Topics Group: 17-21 .....	289
<b>25.5</b>	Review: Advanced Topics Group: 22-24 .....	290
<b>26</b>	<b>Appendix</b>	
<b>26.1</b>	Abbreviated Precedence Chart for C++ Operators .....	291
<b>26.2</b>	C++ Reserved Keywords .....	294
<b>26.3</b>	ASCII Character Set .....	295
<b>26.4</b>	Show Hide File Extensions .....	296
<b>26.5</b>	Academic or Scholastic Dishonesty .....	299
<b>26.6</b>	Successful Learning Skills .....	302
<b>26.7</b>	Study Habits that Build the Brain .....	305
	<b>Glossary</b> .....	306
	<b>Index</b> .....	314
	<b>Attributions</b> .....	318



# Preface<sup>1</sup>

## About this Textbook/Collection

### Programming Fundamentals – A Modular Structured Approach using C++

Programming Fundamentals - A Modular Structured Approach using C++ is written by Kenneth Leroy Busbee, a faculty member at Houston Community College in Houston, Texas. The materials used in this textbook/collection were developed by the author and others as independent modules for publication within the Connexions environment. Programming fundamentals are often divided into three college courses: Modular/Structured, Object Oriented and Data Structures. This textbook/collection covers the first of those three courses.

On January 10, 2013 Version 1.22 was created with the modules that make up the collection "fixed" to their current versions. This will allow Version 1.22 to remain static with the modules as of that date.

The collection is going to be revised with a different organization of chapters along with updated modules to handle C++, Java and C# programming languages. The next version of the collection will have significant changes.

### Connexions Learning Modules

The learning modules of this textbook/collection were written as **standalone** modules. Students using a collection of modules as a textbook will usually view its contents by reading the modules sequentially as presented by the author of the collection.

However, the majority of readers of these modules will find them as a result of an Internet search. The Connexions Project allows the author of a module to create web links to other Connexions modules and Internet locations. These links are shown when viewing materials on-line and are categorized into three types: Example, Prerequisite and Supplemental. The importance of each link is numbered from 1 to 3 by the author. When viewing the module each link shows a three part box with yellow or white rectangles. All three yellow means it is a **strongly related link**. As the yellow decreases the importance decreases.

Students using this collection for a college course should note that all of the **Prerequisite links** within the modules will be modules that student should have already read and most of the **Supplemental links** will be modules that the student will read shortly. Thus, students should use Prerequisite links for review as needed and not be overly concerned about viewing all of the Supplemental links at the first reading of this textbook/collection.

### Conceptual Approach

The learning modules of this textbook/collection were, for the most part, written without consideration of a specific programming language. In many cases the C++ language is discussed as part of the explanation of the concept. Often the examples used for C++ are exactly the same for the Java programming language. However, some modules were written specifically for the C++ programming language. This could not be

---

<sup>1</sup>This content is available online at <<http://cnx.org/content/m19049/1.17/>>.

avoided as the C++ language is used in conjunction with this textbook/collection by the author in teaching college courses.

## Bloodshed Dev-C++ 5 Compiler/IDE

This open source compiler/IDE (Integrated Development Environment) was used to develop the demonstration source code files provided within the modules of this textbook/collection. The compiler/IDE is presented to the student in the second module of Chapter 1, with instructions for downloading, installing and using the compiler/IDE. A more complete explanation of the IDE along with demonstration source code listings with errors is presented in first module of Chapter 5. All of the source code files provided in this textbook/collection contain only ANSI standard C++ code and should work on any standard C++ compiler like Microsoft Visual Studio (which includes C++), Microsoft Visual C++ Express or Borland C++ Builder.

## Instructor Materials

Encrypted instructor materials are available in a module that is not part of this collection. It's title: Instructor Materials for: Programming Fundamentals - A Modular Structured Approach using C++ is available at: <http://cnx.org/content/m34529/latest/><sup>2</sup> and the encryption code is only available to educational institutional faculty that are considering adoption of this collection as a textbook.

## About Connexions

### Connexions Modular Content

The Connexions Project <http://cnx.org><sup>3</sup> is part of the **Open Educational Resources (OER)** movement dedicated to providing high quality learning materials free online, free in printable PDF format, and at low cost in bound volumes through print-on-demand publishing. This textbook is one of many collections available to Connexions users. Each **collection** is composed of a number of re-usable learning modules written in the Connexions XML markup language. Each module may also be re-used (or 're-purposed') as part of other collections and may be used outside of Connexions.

### Re-use and Customization

The Creative Commons (CC) Attribution license<sup>4</sup> applies to all Connexions modules. Under this license, any **Connexions module** may be used or modified for any purpose as long as proper attribution to the original author(s) is maintained. Connexions' authoring tools make re-use (or re-purposing) easy. Therefore, instructors anywhere are permitted to create customized versions of this textbook by editing modules, deleting unneeded modules, and adding their own supplementary modules. Connexions' authoring tools keep track of these changes and maintain the CC license's required attribution to the original authors. This process creates a new collection that can be viewed online, downloaded as a single PDF file, or ordered in any quantity by instructors and students as a low-cost printed textbook.

### Read the book online, print the PDF, or buy a copy of the book.

To browse this textbook online, visit the collection home page. You will then have three options.

1. You may view the collection modules on-line by clicking on the "Start >>" link, which takes you to the first module in the collection. You can then navigate to the next module using "NEXT >>" and

---

<sup>2</sup>"Instructor Materials for: Programming Fundamentals - A Modular Structured Approach using C++"  
<<http://cnx.org/content/m34529/latest/>>

<sup>3</sup><http://cnx.org/>

<sup>4</sup><http://creativecommons.org/licenses/by/2.0/>

through the subsequent modules by using the " $\ll$  PREVIOUS | NEXT  $\gg$ " button that is towards the upper right to move forward and backward in the collection. You can jump to any module in the collection by clicking on that module's title in the "TABLE OF CONTENTS" box on the left side of the window. If these contents are hidden, make them visible by clicking on the small triangle to the right of the "TABLE OF CONTENTS". Chapters also have a small triangle to show or hide contents.

2. You may obtain a PDF of the entire textbook to print or view offline by clicking on the "Download PDF" link in the "Content Actions" box.
3. You may order a bound copy of the collection (for a reasonable printing and shipping fee) by clicking on the "Order printed copy" button.

## Connexions PDF Conversion Problems

Buying a copy of the textbook/collection is basically sending the PDF file to a printing service that has a contract with the Connexions project. There are several known printing problems and the Connexions Project is aware of them and seeking a solution. In the mean time, be aware that quirks exist for printed PDF materials. A description of the known problems are:

1. When it converts an "Example" the PDF displays the first line of an example properly but indents the remaining lines of the example. This problem occurs for the printing of a book (because it prints a PDF) and downloading either a module or a textbook/collection as a PDF.
2. Chapter numbering has been added to the on-line Table of Contents. This will make it easier for students to quickly get to the chapter reading materials. However this creates a "double" chapter numbering within the textbook/collection's PDF and custom printing formats.
3. Within C++ there are three operators that do not convert properly to PDF format.

decrement	-	which is two minus signs
insertion	$\ll$	which is two less than signs
extraction	$\gg$	which is two greater than signs

Table 1

## Rating Connexion Modules

A rating feature was added during 2009 for Connexions modules. It will not be useful until more people rate modules within the Connexions repository. If a module is rated by several people, it can be used as a **measure of quality**. Thus, your participation in rating modules is welcomed and helps others determine the quality of the educational materials being viewed.

In order to rate modules you must have a Connexions account. Three (3) modules have been added to the preface series of modules for this collection. They explain why and how to create a Connexions account and how to rate a Connexions module.



# Author Acknowledgements<sup>5</sup>

I wish to acknowledge the many people who have helped me and have encouraged me in this project.

1. Mr. Abass Alamnehe, who is a fellow faculty member at Houston Community College. He has encouraged the use of Connexions as an "open source" publishing concept. His comments on several modules have led directly to the improvement of the materials in this textbook/collection.
2. The hundreds (most likely a thousand plus) students that I have taken programming courses that I have taught since 1984. The languages include: COBOL, main frame IBM assembly, Intel assembly, Pascal, "C" and "C++". They have often suggested that I write my own book because they thought that I was explaining the subject matter better than the author of the textbook that we were using. Little did my students understand that directly or indirectly they aided in the improvement of the materials from which I taught as well as improving me as a teacher.
3. To my future students and all those that will use this textbook/collection. They will provide suggestions for improvement as well as being the thousand eyes identifying the hard to find typos, etc.
4. My wife, Carol, who supports me in all that I do. She has tolerated the many hours that I have spent in concentration on developing the modules that comprise this work. Without her support, this work would not have happened.

---

<sup>5</sup>This content is available online at <<http://cnx.org/content/m19055/1.5/>>.



# Orientation and Syllabus<sup>6</sup>

## Orientation

### Textbook/Collection Layout

The approach of this course will be to take the student through a progression of materials that will allow the student to develop the skills of programming. The basic unit of study is a Connexions module. Several modules are collected into a chapter. The chapters are divided into five groups.

Group Title	Chapters	Modules
Pre-Chapter Items	N/A	4
Foundation Topics	1-5	27
Modular Programming	6-9	17
Structured Programming	10-16	30
Intermediate Topics	17-21	17
Advanced Topics	22-24	11
Review Materials	N/A	5
Appendix	N/A	7
<b>Total Modules</b>	<b>N/A</b>	<b>118</b>

**Table 1**

Some professors using this textbook/collection might decide to eliminate certain modules or chapters. Some may eliminate the entire Advanced Topics group. Other professors may choose to add additional study materials. The advantage of this textbook/collection is that it may be adapted by professors to suit the needs of their students.

### Chapter Layout

Each chapter will usually flow from:

1. One or more Connexions modules built for independent delivery.
2. A Connexions Practice module built specifically for this textbook/collection.

As you proceed with the Connexions modules that comprise a chapter, you should:

- Complete any tasks/demos that require downloading items.

---

<sup>6</sup>This content is available online at <<http://cnx.org/content/m19684/1.12/>>.

Available for free at Connexions <<http://cnx.org/content/col10621/1.22>>

- Do any exercises.
- Create 3x5 study cards for all definitions. When this material is used as a textbook for a course the definitions are to be memorized. Confirm this with your professor.

As you start the Practice module you will usually encounter:

- Learning Objectives
- Memory Building Activities aka MBAs Link – These could consist of any of the following types of interactive computer activities: flash card, crossword puzzle, seek a word, drag n drop, labeling, ordering or sorting. When the materials are used as a textbook for a course, it is imperative that students do a variety of repetitive activities in order to memorize basic course material. Besides, have fun learning.
- Exercises – In addition to any exercises within the study modules that you completed before the practice module, there will be at least one exercise for students to complete.
- Miscellaneous Items – These will exist for some of the chapters.
- Lab Assignment – Usually, completed on one's own efforts. Review the instructions/restrictions from your professor/teacher if using this for a high school or college credit course.
- Problems – The intent of this activity is for students to formulate their own answers. Thus, solutions to the problems will not be provided. When the materials are used as a textbook for a course, the professor/teacher may assign students to a "Study Group" or let students form study groups to discuss their solutions with each other. If you are using this for a high school or college credit course, verify that you may work as team at solving the problems. This type of approved activity is called "authorized collusion" and is not a violation of "Academic or Scholastic Dishonesty" rules.

A professor using this textbook/collection/course will most likely have additional lab assignments, quizzes and exams that would be used in calculating your grade.

## Connexions Module Reading List

The modules in this textbook/collection have had content reviewed and are believed to be sufficient, thus **no additional textbook is required**. However, some students desire additional references or reading. The author has used several textbooks over the years for teaching "COSC1436 – Programming Fundamentals I" course at Houston Community College and at the Community College of Qatar. A reading reference list has been prepared and includes references for the following textbooks:

1. Starting Out with C++ Early Objects, by: Tony Gaddis et. al., 7<sup>th</sup> Edition, International Edition, ISBN: 978-0-13-137714-1
2. Starting Out with C++ Early Objects, by: Tony Gaddis et. al., 6<sup>th</sup> Edition, ISBN: 0-321-51238-3
3. Starting Out with C++ Early Objects, by: Tony Gaddis et. al., 5<sup>th</sup> Edition, ISBN: 0-321-38348-6
4. Computer Science – A structured Approach using C++, by: Behrouz A. Forouzan et. al., 2<sup>nd</sup> Edition, ISBN: 0-534-37480-8

These textbooks are typically available in the used textbook market at a reasonable price. You may use any one of the three books. If you acquire one of the above **optional** traditional textbooks, you may want to download and store the following file to your storage device (disk drive or flash drive) in an appropriate folder.

Download from Connexions: [Connexions\\_Module\\_Reading\\_List\\_col10621.pdf](#)<sup>7</sup>

## Syllabus

The syllabus for a course that is for credit will be provided by your specific course professor. If you are using this textbook/collection for non-credit as self-study, we have some suggestions:

<sup>7</sup>See the file at [http://cnx.org/content/m19684/latest/Connexions\\_Module\\_Reading\\_List\\_col10621\\_V2.pdf](http://cnx.org/content/m19684/latest/Connexions_Module_Reading_List_col10621_V2.pdf)



1. Plan regular study periods
2. Review the three (3) Pre-Chapter Items modules
3. Review the last four (4) modules in the Appendix
4. Proceed with Chapter 1 going through all 24 chapters
5. Do all of the demo programs as you encounter them
6. Memorize all of the terms and definitions
7. Do all lab assignments
8. Prepare answers to all of the problems in the Practice modules
9. At the end of every section, do the Review module

There is no magic way to learn about computer programming other than to immerse yourself into regular study and **study includes more than casual reading**. To help you keep track of your study, we have included a check off list for the textbook/collection.

Check	Description	# Modules
	Pre-Chapter Items	4
	Last four Appendix Items	4
	Chapters 1 to 5	27
	Review Materials for 1 to 5	1
	Chapters 6 to 9	17
	Review Materials for 6 to 9	1
	Chapters 10 to 16	30
	Review Materials for 10 to 16	1
	Chapters 17 to 21	17
	Review Materials for 17 to 21	1
	Chapters 22 to 24	11
	Review Materials for 22 to 24	1
	First three Appendix Items	3
<b>N/A</b>	<b>Total Modules</b>	<b>118</b>

**Table 2**

## Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

