

# Computer Science Track

## Algorithms in C/C++

<http://cs.stormingrobots.com>

### Table of Content

<u>PREFACE</u> .....	P.1
CHARACTERISTICS ..	2
REQUIREMENT FOR LEVELS PROMOTION ..	3
BOOKS ..	3
SOFTWARE ..	3
HONOR CODE ..	4
<u>ELIGIBILITY IN PARTICIPATING COMPETITIONS</u> ..	4
WHY C ..	5
<u>LEVEL B</u> .....	P.6
<u>LEVEL I</u> .....	P.8
<u>LEVEL II</u> .....	P.10
<u>IF YOU DECIDE TO SELF-STUDY AP CS</u> .....	P.12
<u>LEVEL III</u> .....	P.13
<u>LEVEL IV</u> .....	P.15

# Overview



## Preface

Computer Science skill should go beyond just programming itself but much more importantly in problems solving ability with computational thinking even for grade schools.

Automation is entrenched in our daily lives in the era of the digital age. Computer Science with computational thinking is indispensable for strengthening the foundation. This sub-group aims to build this core foundation and sharpen students' problem-solving skills in this digital world, no matter whether in engineering, or even liberal arts area.

Storming Robots utilizes Robotics to animate problem-solving effort starting in Grade 4. However, we encourage students to study in this Algorithms in C/ C++ Track starting from Grade 8.

This document lists the Syllabus covered in this Track. This syllabus consists of four levels. The levels will be comparable up to college level sophomore or even junior year topics in CS, such as data structure, introduction to algorithms with combinatorial optimization, and complexity analysis.

### Characteristics (important to read first)

- 1) Focus on problems solving/software development skill, so students will not work with the physical robot.
- 2) All levels stress in Computational Thinking and Efficiency.
- 3) Progress is self-paced. All assigned exercises should be completed with excellent quality, with efficiency, memory consumption, readability, maintainability, etc.
- 4) Students will be allowed to move quickly to the next concept only after they demonstrate satisfactory level of understanding through their homework and/on pop-quiz in class. Those with gaps in their prerequisite knowledge will receive additional exercises to address the

shortcomings.

- 5) Allow students to embed other competitions in-between levels.
- 6) Students completed level I with high proficiency will have the capacity to self-study for Advanced Placement Computer Science (AP CS)
- 7) Many Level II students participate the USA Computing Olympiad (USACO) online exam as performance gauges. Do note USACO is far more demanding in terms of analytical, and computational programming skill than AP CS.

### Requirement for Level Promotion:

- 1) This track is broken down into four levels. Students must demonstrate they **can complete all givens exercises and make all corrections required.**
- 2) Pop-quiz problems may be assigned to students to complete in class in order to prove their understanding.

### Books for:

- 1) **Level B, I, & II—C** : C Programming: A Modern Approach, **2nd Edition** by K.N. King - ISBN-13: 978-0393979503, or 978-0393979503.
  - Wish to self-study AP CS? [Review the note in the Level II section.](#)
- 2) **Level III** : . Option : Mastering Algorithms with C: ISBN-13: 978-1565924536, or ISBN-10: 1565924533.
- 3) **Level IV**:
  - Data Structures and Algorithm Analysis in C++, 4th Edition by Mark A. Weiss—ISBN-10: 0273769383.
  - (Option for reference) : C++ Primer (5th Edition) by y Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo. ISBN-10: 0321714113.

### Software:

- ∞ (Windows OS) Eclipse or Microsoft Visual Studio Community Version.
- ∞ (Apple OS) Eclipse, Or XCode. **Please do note that we shall not be able to offer installation advice if you use Apple OS.**

### Honor Code

Programming assignments are pledged work and are bound by the honor code. To simply put, the violation may be in any of the following forms:

- ∞ Claim someone else's work as their own.
- ∞ Edit someone else's work by simply changing variables or style, etc., and claim to be the result of your own work.
- ∞ Complete assigned work with so much substantial help from others that you cannot even explain your own work; but still claim it is the result of your own work.

If found honor code is violated, parents will be informed. Students will not be allowed to participate in any competitions with Storming Robots. With some cases, students will not be accepted back to SR programs.

## Eligibility in participating Competitions/Exams/ Advanced Projects

✓ : must be proficient ①	Minimum Required Levels				
	B	I	II	III	IV
② USACO Silver		✓			
USACO Gold				✓	
USACO Platinum					✓
RCJ Line	✓				
RCJ Maze				✓	
RCJ Soccer		✓			
③ ZeroRobotics		✓			
④ Advanced Projects				✓	
⑤ C Programming Certified Associate			✓	✓	
⑤ C ++ Programming Certified Associate				✓	





- ① It indicates minimum required level. Higher level is a must in order to do well in the competition or exam.
- ② USACO is an online exam. It is designed to be challenging even for the best students. E.g. Students scored "5" in AP CS may not do well in even Bronze level (the level below Silver).
- ③ Require minimum Pre-Cal & Algebra II. Linear Algebra will be helpful.
- ④ See more details at : <https://mp.stormingrobots.com>
- ⑤ Certification from C++ Institution with Cisco Networking Academy .

To obtain more details about these activities: <https://compete.stormingrobots.com>.

## Why C ?






[The Top Programming Languages 2017 Survey by IEEE.](#)

### Rank within Embedded Systems realm

Language Rank	Types	Spectrum Ranking
1. C		99.7
2. C++		97.2
3. Arduino		73.0
4. Assembly		72.1

Note: Arduino is a platform having its own set of APIs, not a programming language. The IEEE article included it because Arduino has become the basis of a large volume of prototype devices.

### Rank by Overall

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		99.7
3. Java		99.4
4. C++		97.2
5. C#		88.6

## Why C over Python

We focus on educating students to “understand” the underlying structure, think efficiency and analysis with algorithms; not to learn a specific language. This will equip students the skill to adapt the mechanics of new modern tool requiring various algorithms and optimization techniques.

C programming allows high flexibility and the power to perform closer to the machine than modern languages. It is still often used in embedded systems or high-performance computing. It is considered to be the lingua franca of programming.

Python is designed for Data Scientist to program at ease so that they can focus on a specific subject matter. While Python is definitely a great tool and easy to code, using it as primary programming language to build sound foundation in algorithmic analysis is counterproductive to our goal.

# Level B



This covers the bare fundamentals of computer programming from basic expressions to compound control structure.

## Learning Outcome

- ∞ By the time this level is completed, students should have completed 20+/- programs up to Chapter 6 from the K.N. King book. This is not a hard-set number because more exercises will be given if necessary in order to strengthen one's understanding in a certain subject matter; or vice versa.
- ∞ Possess the knowledge in the fundamentals of programming with an emphasis on producing clear, robust, and reasonably efficient code using top-down design, informal analysis, and effective testing and debugging.
- ∞ Mindset in computational thinking and analysis.
- ∞ Know how to use Debugger for trouble shooting.
- ∞ At completion of Level B, students should be able to do simple control structure, take a straight forward problem set to resolve it with computer programming in C.
- ∞ Basic mechanic understanding in utilizing Array structure.
- ∞ Exploratory usage of a “struct” type.

If you join this class with prior programming background in RobotC Robotics Projects I & II Analytics, it will help you to accelerate through the first four chapters.

## Book :

C Programming: A Modern Approach, 2nd Edition by K.N. King - ISBN-13: 978-0393979503, or 978-0393979503.

## What will be covered

1. Introducing C and Strengths and Weaknesses of C - Chapter 1
2. How to use the Microsoft Visual C/C++ Compiler IDE.
3. C Fundamentals in writing Simple Program - Chapter 2
- 1) Standard Formatted Input/Output - Chapter 3
- 2) Expressions (simple to compound) - Chapter 4
- 3) Rudimentary level in using functions() – class note
- 4) Selection/ if - Control Statements | Boolean Expressions - Chapter 5
- 5) Introduction of State Tables/Diagrams. Simple idea in State machine—more example exercises using “switch” to implement state tables. More in depth will be covered in Later Levels.
- 6) Loops Control Structure - Chapter 6
- 7) Further understanding in data types, signed vs. unsigned, further reinforcement in Loops Control Structure - Chapter 7
- 8) Basic understanding in Array Type. Class note and Partial Chapter 8. Will focus only on the mechanic in using array.
- 9) Basic understanding in “struct” type. Class note and Partial Chapter 16.
- 10) Creating Functions (Partial Chapter 9).
- 11) Know the basics in Windows File system.



## To conclude Level B

Summary of the chapters covered in this level - Chapter 1 to partial 8.

All final work should follow [prescribed Programming Style](#) .

There will additional assignments outside the book given to improve computational skill with simple algorithms such as Sieve of Eratosthenes, simple Euclid Algorithm, Primality Test, etc.

There will be pop quiz given at random time to students to ensure satisfactory proficiency in the covered concepts. If students fail to perform well for the pop quiz, he/she will not be allowed to promote to the next level.

# Level I

## - Data Structure 1 (C)-



All students who complete this level should participate the Bronze level of [USACO](#) online exams. USACO contests are designed to challenge even the very best students.

For the students who have taken AP CS from high school will still find this level challenging. Completion of Level I is required for all students who wish to participate in any Open level of Robotics Competition.

### Learning Outcome:

- ∞ *Should be able to tackle Bronze, or even Level in the USACO. Based on our history, all students who demonstrate high proficiency in completing Level I and have done some practices have promoted to at least Silver Level. Some even got Gold.*
- ∞ *You not only should have covered the core control structure foundation required in AP Computer Science, but also will be stronger in programming analysis. Most programs exercises you have completed in even Level B under this program requires higher analytical skill over what is required in AP Comp. Sci. The only additional content is in the basics in OOPS design pattern. That will be covered in Level II—OOPS.*
- ∞ Students demonstrated high proficiency in this level will be able to self-study for AP Computer Science—A with great ease.
- ∞ Become more efficiency in Debugger including break points, observation of variables, watch feature.
- ∞ Should expect to work additional exercises outside the book as well .
- ∞ After completion of this, maximum selective top 10 students will be invited to take the C Programming Associate Certification Exam offered by C++ Institution at no cost (\$300 exam otherwise).

### What will be covered

- 1) More advanced analytical work on Arrays : Complete Chapter 8 with additional classwork.



- Heavily focus on Multi- Dimensional
  - Additional analytical work with nested loop
  - Should expect exercises more complex than those in Level B.
- 2) Create functions & process from compilation to linkage - Chapter 9 & 10 (partial).
  - 3) Base Conversion — Base-8, 16, Binary.
    - Utilize the memory addresses shown in the debugger
  - 4) Bitwise operations — Chapter 20
  - 5) Memory pointers with 1D and 2D array – Chapter 11 & 12
    - C string , Arrays of Strings
    - Dynamic allocation.
  - 6) String manipulation — Chapter 13
    - C string functions with focus on pointers arithmetic
  - 7) Struct, Union, Enumeration — Chapter 16.
    - Using bits in structure
    - Review Exercises to summarize Usage of array of struct, enum, and pointers.
  - 8) Basic File I/O — Chapter 22 (with C & C++ - add'l note).
  - 9) Simple Command-Line Operations (Chapter 15 and add'l note).
    - Compile and link programs without using the IDE
    - Use system path, environment variables, etc.
    - Automate testing with a set of test data files
    - Create your own library.
  - 10) Preprocessor - Chapter 14 —Conditional Compilation, such as #if, #elif, #line, etc. Its role in compilation process.



## To conclude Level I:

- Exercise external to the book may include matrix Multiplication, Prime Sieving, Modular Inverse, Fast Exponentiation, The Euclidean Algorithm, Fibonacci with dynamic programming method.
- Summary of the chapters covered in this level ( not necessary in the this order): Chapter 9,10, 11, 12, 13, 14, 15, 16, and 22.

# Level II

## - Data Structure II (C) -



### Learning Outcome:

- ∞ Further understanding in College level Linear Data Structure.
- ∞ Improvement in modular programming design.
- ∞ Further reinforce algorithmic problems solving skills with computer programming.
- ∞ By the time you complete this level, you are already equipped with very sound programming skill comparable to most undergrad CS capacity.
- ∞ During November and April, students are encouraged to continue working on USACO problems set, as well as take the Bronze to Silver Level online Exam.

### Taking AP CS?

*We do not offer classes during school year focusing on AP CS because of its simplicity. Our Level II students will simply self-study Java with great ease instead, and usually within a short period of time. Many simply purchase the AP CS Bar Exam Book for practice.*

**DO NOTE:** *If you have yet finished Level II with SR but wish to study for AP CS, you should pursue a full-blown Java course. You are recommended to take our JAVA AP CS Summer Workshop, or one with other institution.*

### What will be covered

- 1) Recursion
  - stack data structure — more in-depth understanding how it works in memory stack. (beyond coverage of the recursion in Ch-9.
  - Writing simple Recursive functions with ease, including Fibonacci with basics in dynamic programming method.
  - Math Expression Parsing (postfix)
  - Tower of Hanoi

- 2) Basic in Binary Search (Divide and Conquer Algorithm)
- 3) Commonly used sorting algorithms :
  - Insertion sort
  - Quicksort (Divide and Conquer Algorithm). With int / char ar-
- 4) Explore Deterministic Finite State Machine | Flowchart
- 5) Advanced Uses of Pointers — Chapter 17 and class note
  - Polymorphism in C
  - Pointer to function, such as callbacks like in Quick Sort.
  - Pointer to Pointer
  - Generic programming with void\* ( Quick sort using void \* )
  - Self-contained Structure Abstraction, such as with array of pointers to function, nested structure.
  - Variable list of arguments.
- 6) Storage Classes— Chapter 18
- 7) Basics in Abstract Data Type – Chapter 19
  - Linked List— single | double | circular . (Explore simple tree structure)
  - Basic Stack (LIFO)— Push / Pop
- 8) System Signal (interrupt) - Chapter 24



## To conclude Level II:

- **This level will conclude the usage of the Dr. K.N. King’s Book.** Students should skim thru Ch. 21, 23 to 25-27 on their own, as they all should be used for reference and self-study purpose only.
- Most exercises in this level are external from the book. Many of them take a more pragmatic approach to emphasize its application and analysis, and require much beyond just knowing the mechanic of coding.
- May Practice on several USACO Silver and/or Gold level exercises depending on the timeframe and students’ analytical skills.

# If you decide to self-study AP CS

**DO NOTE:** If you have yet finished Level II with SR but wish to study for AP CS, you should pursue a full-blown Java course. You are recommended to take SR's JAVA AP CS Summer Workshop, or one with other institution during the school year.

Although the AP CS is simple, you should dedicate time to work on all the topics in the AP Baron Practice Book. According to our level II students, they spent only three months to complete the AP Practice book (some said a few weeks), and still scored “5” from exam.

HOWEVER, if you have to spend time on learning Java, you should gain good level of proficiency. The goal is not just to get “5” from a simple exam, but gain reasonable proficiency. The following list recommended topics and exercises, some are beyond AP CS—A requires.

The following list is what covered by AP CS (from 2016). You should always consult the college board AP CS website for the latest update.

## Classes and Objects

- Classes
- Declaring Classes
- Declaring Member Variables
- Defining Methods
- Providing Constructors for Your Classes
- Passing Information to a Method | Constructor
- Objects—Creating Objects
- Using Objects
- Returning a Value from a Method
- Using the this Keyword
- Controlling Access to Members of a Class

## Enumeration Type (not in AP CS)

- Numbers and Strings Wrapper Classes
- The Numbers Classes
- Formatting Numeric Print Output
- Characters
- Strings
- Converting Between Numbers and Strings
- Manipulating Characters in a String
- Comparing Strings and Portions of Strings

## Interfaces

- Defining and Implementing an Interface
- Using an Interface as a Type
- Evolving Interfaces
- Default Methods

## Inheritance

- Multiple Inheritance of State, Implementation, and Type
- Overriding and Hiding Methods
- Polymorphism (somewhat in AP CS)
- Hiding Fields (not in AP CS)
- Using the Keyword super
- Object as a Superclass
- Writing Final Classes and Methods
- Abstract Methods and Classes

Search : Sequential and Binary Search

Sorting: Selection | Insertion | QuickSort | Merge-Sort (knowing the mechanic, not implementation)

[Handling Exceptions](#) (not in AP CS)

# Level III

## ADT with Algorithms in C



### Learning Outcome:

Students will have covered the most commonly discussed algorithms / techniques used in many problems which require performance. Students who are able to complete this level will have proven themselves excellence in not just programming, but also in computational problem-solving. All of the students who can reach this level should gain a highly competitive edge in college internship programs. Many of our alumni reported to us that they got internship interview even at College Freshmen year.

### Books:

- 1) **Mastering Algorithms with C:** Useful Techniques from Sorting to Encryption. ISBN-13: 978-1565924536, or ISBN-10: 1565924533. This book will be used for Level III with additional notes.
- 2) (Option for reference) : Data Structures and Algorithm Analysis in C++, 4th Edition by Mark A. Weiss—ISBN-10: 0273769383.
- 3) (Option for reference) : C++ Primer (5th Edition) by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo. ISBN-10: 0321714113.

### What will be covered

This may be modified depending on students' interest and/or possible independent project. Most topics are from the Mastering Algorithms in C Book

- 1) Introduction to Data Structures and Algorithms —Ch. 1
- 2) Advanced Pointer Manipulation — Ch. 2
- 3) Analysis of Algorithms — Ch.4
- 4) Review DFS using recursion with:
  - Minimax and Alpha Beta Pruning (DFS | Tree Structure | Basic Backtracking Algorithm) (Class Note).
  - Backtracking NQueen Algorithm

- 5) More on Linked List—Ch. 5
  - Utilize C++ Vector | List | Map
- 6) More in queues —Ch. 6
  - Event Handling
  - Breath First Search
  - Work with maze project
- 8) Huffman Coding (Loss-less compression | Tree Structure)
- 9) Maze Navigation (BFS)
  - This may be done in a maze setting with two possible challenge: Traverse the whole maze to seek for location of a certain target. BFS to go back to the start point to report the location of the targets
- 10) Knapsack Algorithm (including items and using dynamic programming).
- 11) Explore Sets and Graphs—Ch. 7 & 1
- 12) Hashing—Ch.8
  - Hash Table - Linear Probing, Open addressing, Collision avoidance
- 13) Trees—Ch.9
  - Explore Recursive Binary Tree Algorithms — Class Notes
  - Review Self-balance BinaryTree (AVL)
- 1) Heaps and Priority Queue—ch.10
- 2) Numerical Methods—Ch. 13
- 3) Data Compression—Ch. 14 (Greedy Algorithm)
  - Huffman Coding Algorithm
- 4) Geometric Algorithms—Ch. 17
  - Convex Hulls

# Level IV

## (Advanced Algorithms using C++)



By the time this level is completed, students should have a sound grasp of Object-Oriented Design Pattern. If you are going into computer engineering or any courseware which involves working with micro-controller libraries, completion of this will help you greatly in working with these libraries.

Students will gain the advantage of using Standard Library (STL) Abstractions to reduce complexity of their own program solutions, but still understanding the C++'s extra features coming with overhead. Besides, C++ exemplifies the usage of abstract data structure with reduced amount of low level detailed work as well. This is designed for students who are interested in Robotics Engineering and AI learning; both realms require high performance. Most resources / libraries written in high level languages for embedded systems are in either C++ or C.

C++ is a superset of C, it means it will contain many important features which will reduce your work load vs using just C, especially in arrays, list, string, memory management.

### Learning Outcome:

If students possess excellent “Analytical” math level, they should do well in obtaining Gold level for [www.usaco.org](http://www.usaco.org). However, nothing can replace with the passion of “problem solving” and “Practice!”

### Books:

- ∞ Mastering Algorithms with C: Useful Techniques from Sorting to Encryption. ISBN-13: 978-1565924536, or ISBN-10: 1565924533.
- ∞ Data Structures and Algorithm Analysis in C++ : Edition 2013 . ISBN-13: 978-0273769385, or ISBN-10: 0273769383

## Covered Topics from the C++ Data Structure Book

- 1) Math that you need to know in Algorithm Analysis — Partial Ch. 1 & 2
- 2) Basics in OOPS structure— Partial Ch. 1 and class notes
- 3) Further investigate into ADT —Ch. 3
- 4) String vs. Array —( Ch 1 & 2 in the C++ Primer and class notes.)
- 5) Reference Pointers Type, and Exception Handling— (Ch. 4 in the C++ Primer)
- 6) Vectors | List | Deque— ( Ch 3 in the C++ Primer and class notes.)
- 7) Sequential containers (Iterators) and strings—(Ch. 5 in the C++ Primer)
- 8) Sets | Maps—Ch. 4  
Performance on using `map< >`— (Ch. 7 in C++ Primer)
- 9) Automatic Balanced Tree—Ch. 4
- 10) Revisit Hashing without Linked list using STL—Ch. 5  
Linear and Quadratic Probing and ReProbing  
Collision Avoidance
- 11) Sets | Disjointed Sets.—Ch. 8  
Minimum Spanning Tree Implementation.
- 12) More in Dynamic Programming Technique:
  - 1) With array table
  - 2) With ordering matrix multiplications
  - 3) Optimal binary search

\*\*\* Note : Many topics in the C++ Primer Book should be very simple to review if you have already completed Level II work at SR. Topics that you can review on your own in the C++ Primer Book:

- 1) Some commonly used methods in Algorithm library—Ch. 6
- 2) Generic Programming —Ch. 8
- 3) Virtual | Friend Types —Ch. 15



## Algorithms covered

### Greedy / Graph / Tree Algorithms:

- |  |                            |
|--|----------------------------|
| 1) Maze Navigation (Breath First Search / Queuing Structure) | 10) LIS and LCS            |
| 2) MinMax & Alpha-Beta Pruning                               | 11) Job Scheduling Problem |
| 3) Huffman Coding Compression                                | 12) Genetic Programming    |
| 4) Dijkstra's Shortest path                                  | 13) Lempel and Ziv (LZ78)  |
| 5) Travelling Salesman Problem                               | 14) SHA-1                  |
| 6) A* algorithm  | 15) Option : LZW           |
| 7) Kruskal's Minimal Spanning Tree                           |                            |
| 8) Prim's Minimal Spanning Tree                              |                            |
| 9) Genetic Search Algorithm                                  |                            |

### Divide and Conquer:

- 1) Merge Sort
- 2) Quick Sort
- 3) Binary Search

### Combinatorial optimization

- 1) Fibonacci number series with dynamic programming
- 2) Knapsack with items and dynamic programming

### Others (Tentative):

- 1) Parallel programming with semaphore.
- 2) Floyd Warshall Algorithm

### Pattern Recognition

- 1) Use Raspberry PI—Linux Platform
- 2) Nearest—Neighbour (K-NN)
- 3) Canny-Edge

