XML Design Editor

<https://labs.udacity.com/android-visualizer/#/android/text-view>

Basic Android Concepts

<https://classroom.udacity.com/courses/ud834-india/lessons/4027328704/concepts/caca432b-c83a-4740-8109-95742d7d4409>

Take Android App Screenshot from Logcat

<https://developer.android.com/studio/debug/am-screenshot.html>

Google Design Rules

<https://material.io/design/typography/#applying-the-type-scale>

<https://design.google/resources/>

<https://material.io/tools/theme-editor/>

Material Palettes

<https://www.materialpalette.com/orange/yellow>

<https://paletton.com/>

Android Vocabulary

<https://developers.google.com/android/for-all/vocab-words/?hl=en>

Recommended Android Course on Udemy

<https://www.udemy.com/the-complete-android-oreo-developer-course/>

Developer Guide Android By Google

<https://developer.android.com/guide>

UX Flow

<http://wireflow.co/>

<https://uxdesign.cc/when-to-use-user-flows-guide-8b26ca9aa36a>

[**https://bilgisayarkavramlari.com/2009/11/23/arama-algoritmalari-search-algorithms/**](https://bilgisayarkavramlari.com/2009/11/23/arama-algoritmalari-search-algorithms/)

[**https://bilgisayarkavramlari.com/2008/08/09/siralama-algoritmalari-sorting-algorithms/**](https://bilgisayarkavramlari.com/2008/08/09/siralama-algoritmalari-sorting-algorithms/)

**------------------------------------------------------------------------------------------------------------------------------------**

**CS106A - Programming Methodology**

<https://see.stanford.edu/Course/CS106A>

<https://cs.stanford.edu/people/eroberts/courses/cs106a/lectures/index.html>

<https://web.stanford.edu/class/archive/cs/cs106a/cs106a.1142/lectures/>

**------------------------------------------------------------------------------------------------------------------------------------**

**C Programming Tutorial**

(\*) <http://www.btechsmartclass.com/c_programming/introduction-to-c-programming.html>

(\*) <https://www.programiz.com/c-programming>

<https://codeforwin.org/2017/08/introduction-c-programming.html>

<https://overiq.com/c-programming-101/>

<https://www.geeksforgeeks.org/c-programming-language/>

<https://www.tutorialspoint.com/cprogramming/index.htm>

<https://www.learn-c.org/>

<https://www.learnc.net/>

<https://www.tenouk.com/download.html>

<https://cs50.harvard.edu/x/2021/>

<https://users.cs.cf.ac.uk/Dave.Marshall/C/>

<http://www.cs.cornell.edu/courses/cs113/2006fa/Write_Your_First_C_Program.html>

**------------------------------------------------------------------------------------------------------------------------------------**

**C++ Programming Tutorial**

(\*) <http://www.btechsmartclass.com/cpp-programming/>

(\*) <https://www.programiz.com/cpp-programming>

<https://www.geeksforgeeks.org/c-plus-plus/>

<https://www3.ntu.edu.sg/home/ehchua/programming/index.html#Cpp> (also for C)

**------------------------------------------------------------------------------------------------------------------------------------**

**Complier, Assembler, Linker and Loader**

<https://sites.google.com/site/kmrvikash/home/tutorials/c-tutorials/compiler-assembler-linker-and-loader-a-brief-story>

**------------------------------------------------------------------------------------------------------------------------------------**

**Brief Information About Executable Load Files (ELF) and ASM Codes**

<https://en.wikipedia.org/wiki/Executable_and_Linkable_Format> (Executable and Linkable Format)

<http://binvis.io/#/>

<https://gist.github.com/mikesmullin/6259449> (Mike's x86-64 Assembly (ASM) Notes)

<https://www.cs.uaf.edu/2016/fall/cs301/lecture/09_28_machinecode.html> (Machine Code in x86)

<https://en.wikipedia.org/wiki/X86_instruction_listings> (x86 instruction listings)

**------------------------------------------------------------------------------------------------------------------------------------**

**20 issues of porting 32-bit C++ Code to the 64-bit platform**

<https://pvs-studio.com/en/blog/posts/cpp/a0004/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Cmake Tutorial (for C/C++)**

<https://www.bogotobogo.com/cplusplus/files/cmake/CMake-tutorial-pdf.pdf>

<https://www.cvl.isy.liu.se/en/education/graduate/opencv/CMake%20presentation.pdf>

<https://slideplayer.com/slide/11663654/> (Cmake Tutorial)

<http://derekmolloy.ie/hello-world-introductions-to-cmake/>

<https://frankie-yanfeng.github.io/2019/11/12/CMake-2019/>

<https://mirkokiefer.com/cmake-by-example-f95eb47d45b1>

<https://cmake.org/cmake/help/latest/guide/tutorial/index.html>

<https://cmake.org/runningcmake/>

<https://docs.microsoft.com/tr-tr/cpp/build/cmake-projects-in-visual-studio?view=msvc-160>

<https://www.kitware.com/platforms/> (cmake)

**------------------------------------------------------------------------------------------------------------------------------------**

**Cmake Tutorial (for Java)**

<https://cmake.org/pipermail/cmake/2015-December/062173.html>

<https://github.com/ptitpoulpe/cmake-swig-java-example>

<http://www.swig.org/Doc1.3/Java.html>

**------------------------------------------------------------------------------------------------------------------------------------**

**Make for C and Java**

<https://www.cs.swarthmore.edu/~newhall/unixhelp/howto_makefiles.html>

<https://www.cs.swarthmore.edu/~newhall/unixhelp/javamakefiles.html>

<https://stackoverflow.com/questions/32127524/how-to-install-and-use-make-in-windows>

**------------------------------------------------------------------------------------------------------------------------------------**

**Ant / Maven / Gradle**

<https://www.baeldung.com/ant-maven-gradle>

**------------------------------------------------------------------------------------------------------------------------------------**

**GCC / G++**

<https://www.geeksforgeeks.org/difference-between-gcc-and-g/>

<https://www.geeksforgeeks.org/compile-32-bit-program-64-bit-gcc-c-c/?ref=rp>

**------------------------------------------------------------------------------------------------------------------------------------**

**Lex&Yacc**

<https://www.epaperpress.com/lexandyacc/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Java Programming Tutorials**

(\*) <http://www.btechsmartclass.com/java/java-index.html>

(\*) <https://www.programiz.com/java-programming>

<https://www3.ntu.edu.sg/home/ehchua/programming/index.html#Java>

<http://www.pskills.in/java/index.jsp>

<https://www.geeksforgeeks.org/java/>

<http://indico.ictp.it/event/a0727/session/7/contribution/4/material/0/0.pdf>

<http://indico.ictp.it/event/a0727/session/7/contribution/4/material/0/1.pdf>

**------------------------------------------------------------------------------------------------------------------------------------**

**Jar Distribution**

<https://introcs.cs.princeton.edu/java/85application/jar/jar.html>

<https://www.baeldung.com/java-jar-executable-manifest-main-class>

<https://docs.oracle.com/javase/tutorial/deployment/jar/build.html>

<https://www.geeksforgeeks.org/jar-files-java/>

<https://www.geeksforgeeks.org/working-with-jar-and-manifest-files-in-java/>

**------------------------------------------------------------------------------------------------------------------------------------**

**C# Programming Tutorials**

<https://www.programiz.com/csharp-programming>

<https://www.geeksforgeeks.org/csharp-programming-language/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Markdown Syntax**

<https://www.makeareadme.com/>

<https://www.markdownguide.org/basic-syntax/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Similarities and Differences between Java and C++**

<https://www.geeksforgeeks.org/similarities-and-difference-between-java-and-c/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Developer Knowledge Platforms**

<https://martinfowler.com/>

<https://www.c-sharpcorner.com>

<https://www.codeproject.com>

<https://stackoverflow.com>

<https://www.udemy.com>

<https://www.pluralsight.com>

<https://www.tutorialspoint.com>

<https://www.geeksforgeeks.org>

<http://www.csharpnedir.com/>

<https://www.reddit.com/r/learnprogramming/>

<https://www.reddit.com/r/programming/>

<https://stackoverflow.com/>

<https://serverfault.com/>

<https://techcrunch.com/>

<https://news.ycombinator.com/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Database Management**

<https://www.vertabelo.com/blog/how-to-create-a-database-model-from-scratch/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Operating Systems**

<https://www.geeksforgeeks.org/operating-systems/#basics>

<https://www.geeksforgeeks.org/difference-between-dos-and-windows-2/>

<https://www.geeksforgeeks.org/difference-between-dos-and-windows/>

<https://www.geeksforgeeks.org/difference-between-user-level-thread-and-kernel-level-thread/>

<https://www.geeksforgeeks.org/whats-difference-between-linux-and-android/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Remote Connections**

<https://phoenixnap.com/kb/ssh-to-connect-to-remote-server-linux-or-windows>

<https://cat.pdx.edu/platforms/windows/remote-access/>

<https://medium.com/@sddkal/xrdp-ile-windows-linux-uzak-masa%C3%BCst%C3%BC-ba%C4%9Flant%C4%B1s%C4%B1-c3a50441a9b4>

**------------------------------------------------------------------------------------------------------------------------------------**

**Visual Representation of Algorithms**

<https://visualgo.net/en/sorting>

**------------------------------------------------------------------------------------------------------------------------------------**

**Git**

<https://rogerdudler.github.io/git-guide/index.html>

<https://www.edureka.co/blog/git-tutorial/>

[https://dev.to/lydiahallie/cs-visualized-useful-git-commands-37p1#merge](https://dev.to/lydiahallie/cs-visualized-useful-git-commands-37p1)

<https://dev.to/chrissiemhrk/git-commit-message-5e21>

<https://www.youtube.com/watch?v=MJUJ4wbFm_A&ab_channel=CS50>

**------------------------------------------------------------------------------------------------------------------------------------**

**Host a web site**

<https://pages.github.com/>

<https://www.netlify.com/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Host a web app**

<https://www.heroku.com/platform>

<https://aws.amazon.com/education/awseducate/>

<https://azure.microsoft.com/en-us/free/students/>

<https://edu.google.com/programs/students/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Source Code Examples**

c-color-console

c-console-menu-click

c-donut-deobfuscated

c-nweb-webserver

c-plot-graph

c-progressbar-console

c-qr-console-demo

console-snake-game

AverageSample

LangSample

**------------------------------------------------------------------------------------------------------------------------------------**

**Big-O Cheat Sheet**

<https://www.bigocheatsheet.com/>

<https://rithmschool.github.io/function-timer-demo/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Sorting Algorithms**

https://sorting.at/

<https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>

<https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/visualize/>

<https://visualgo.net/en/sorting>

<https://www.toptal.com/developers/sorting-algorithms>

<https://math.hws.edu/eck/js/sorting/xSortLab.html>

**------------------------------------------------------------------------------------------------------------------------------------**

**Matrix Chain Multiplication**

<https://www.youtube.com/watch?v=JMql7zF87aE>

<https://www.geeksforgeeks.org/matrix-chain-multiplication-dp-8/>

**------------------------------------------------------------------------------------------------------------------------------------**

**LCS**

[https://www.geeksforgeeks.org/longest-common-subsequence-dp-4/](https://www.geeksforgeeks.org/longest-common-subsequence-dp-4/%20)

[https://en.wikipedia.org/wiki/Longest\_common\_subsequence\_problem](https://en.wikipedia.org/wiki/Longest_common_subsequence_problem%20)

<https://handwiki.org/wiki/Longest_common_subsequence_problem>

<https://www.geeksforgeeks.org/printing-longest-common-subsequence-set-2-printing/>

**------------------------------------------------------------------------------------------------------------------------------------**

**Convex Hull**

<https://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/>

<https://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/?ref=lbp>

<https://www.youtube.com/watch?v=B2AJoQSZf4M&ab_channel=StableSort>

**------------------------------------------------------------------------------------------------------------------------------------**

**Greedy Algorithms**

<https://jameskle.com/writes/greedy-algorithm-dynamic-programming>

**------------------------------------------------------------------------------------------------------------------------------------**

* 1. **Knapsack Problem**

<https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/>

<https://www.geeksforgeeks.org/printing-items-01-knapsack/>

<https://www.youtube.com/watch?v=-kedQt2UmnE&ab_channel=StableSort>

<https://www.youtube.com/watch?v=CUAolXf8u-U&ab_channel=StephenO%27Neill>

<https://www.es.ele.tue.nl/education/5MC10/Solutions/knapsack.pdf>

<http://cse.unl.edu/~goddard/Courses/CSCE310J/Lectures/Lecture8-DynamicProgramming.pdf>

**------------------------------------------------------------------------------------------------------------------------------------**

**Cryptographic Hash Functions**

[https://en.wikipedia.org/wiki/Cryptographic\_hash\_function](https://en.wikipedia.org/wiki/Cryptographic_hash_function%20)

[https://en.wikipedia.org/wiki/Cyclic\_redundancy\_check](https://en.wikipedia.org/wiki/Cyclic_redundancy_check%20)

[https://en.wikipedia.org/wiki/MD5](https://en.wikipedia.org/wiki/MD5%20)

[https://en.wikipedia.org/wiki/SHA-1](https://en.wikipedia.org/wiki/SHA-1%20)

[https://en.wikipedia.org/wiki/SHA-2](https://en.wikipedia.org/wiki/SHA-2%20)

**------------------------------------------------------------------------------------------------------------------------------------**

**Algorithms and Programming – I**

**Week-1 (Introduction to Computer Systems)**

1. Course Plan and Communication
2. Grading System, Home works, and Exams.
3. Computer Engineering Job Qualifications
4. Developer Road Map
5. Using Google
6. Programming Introduction
   1. Operating System Basics
      1. Types of Operating Systems
      2. Console commands
      3. System folders
      4. System parameters
      5. Storage management
   2. Computer Network Basics
      1. Network connections
      2. Network related console commands (ipconfig, ipconfig /renew, ipconfig /release, hostname, netstat –a, nslookup)
      3. IP, Port, DNS settings, NAT etc.
      4. Remote connections (FTP, SSH, RDP, XRDP)
      5. Putty, Mobaxterm
   3. Numerical System Basics
      1. Binary system
      2. Hexadecimal system
   4. Character Sets

**Week-2 (Introducton to Algorithms and Development Envoriments)**

1. Algorithm Basics
2. Introduction to Analysis of Algorithms
   1. Algorithm Basics
   2. Flowgorithm
   3. Pseudocode
3. Programming Topics
   1. Programming Environment Setup and Configuration
      1. C / Cpp
         1. DevCpp
         2. Code Blocks
         3. GCC/G++ Compiler (Mingw) / Clang-cl (LLVM)
         4. vscode
         5. Visual Studio Community Edition
            1. Visual Studio x64 x86 Configurations and Features
            2. Project Types
         6. Notepad++
         7. Vi/Vim
         8. Eclipse
            1. Simple Java Project Generation
            2. Jar Export as Library or Executable
            3. Maven Project Generation
            4. Junit Test Case Generation and Testing
         9. Netbeans
         10. Turbo C
         11. Turbo C++
      2. Java
         1. JDK, JRE Setup
         2. System Environments and Paths
         3. Netbeans
         4. Eclipse
         5. Intellij Idea (jetbrains)
         6. vscode
         7. Notepad++
      3. C#
         1. Visual Studio Community Edition
         2. Notepad++
   2. Programming Environment Setup and Configuration
      1. Notepad++ (Notepad for Source Code)
      2. HxD (Hex Editor)
      3. Marktext (Markdown Syntax Editor)
      4. Cygwin (Linux environment for Windows)
      5. Dependency Walker (32-bit or 64-bit Windows module dependency checker)
      6. Doxygen (Code Documentation)
      7. Sonarlint (Code Quality and Code Security Extension)
      8. Codepen.io (online code sharing)
      9. Codeshare.io (real time code sharing)
      10. Codebeautify.org (online data conversion tools)
      11. AsciiFlow.com (ASCII drawing tool)
      12. Freemind (opensource mindmap application)
      13. Wireflow (user flow designer)
      14. PlantUML (software designer)
      15. Drawio (drawing tool)
      16. Putty (Remote Connection)
      17. MobaXterm (Remote Connection)
      18. Teamviewer (Remote Connection)
      19. Paletton.com (Color Chooser)
      20. Understand (Static Code Analysis)
      21. JD Project (Java Decompiler)
      22. Cutter (Multi-Platform Reverse Engineering Tool)
      23. IDA Pro / Freeware (Native Reverse Engineering Tool)
      24. Travis-CI
          1. Travis.yml
      25. Jenkins
      26. Valgrind
      27. Docker
          1. Dockerfile
          2. DockerHub
          3. Docker Compose Yaml
          4. Dockerrun.aws.json (AWS)
      28. Nuget Packages
      29. Extras
          1. vim/vim-wim32-installer (windows vim installer)
      30. SCV Cryptomanager
      31. Addario CryptoBench
      32. Raymond’s MD5 & SHA Checksum Utility
      33. SlavaSoft HashCalc
      34. Portable PGP
   3. Online Programming Envoriments
      1. Hackerrank
      2. CS50 Sandbox
      3. Programiz C Online Complier

**Week-3 (Introduction to Source Code Version Management)**

1. Programming Source Code Sharing and Version Management
   1. Introduction to Source Code Management Systems
   2. Features of Source Code Management
   3. Why Do We Need Source Code Management?
   4. Types of Version Control Systems
      1. Centralized
      2. Distributed
   5. Git Usage
      1. Installation of Git
         1. Git
         2. Git-Extension
         3. Gig (git ignore creator)
      2. Configuration of Git
      3. Github Platform Usage
      4. Create a New Repository
      5. Checkout a Repository
      6. Add & Commit (Write Good Commits)
      7. Pushing Changes
         1. Update Local Repo Before Sending
         2. Send Changes to Remote Repo
      8. Branching
      9. Update & Merge
         1. Fast-forward (-ff) Merging
         2. No-fast-forward (--no-ff) Merging
         3. Merge Conflicts
      10. Rebasing
      11. Replace Local Changes / Resetting
          1. Soft reset
          2. Hard reset
          3. Reverting
      12. Cherry-picking
      13. Fetching
      14. Reflog
      15. Tagging
      16. Log
      17. Gource
   6. Maven Usage
   7. TFS Usage
   8. SVN Usage

**Week-4 (Introduction to Code Reusability and Automated Testing)**

1. Shared Library Development
   1. C Programming
   2. Cpp Programming
   3. Csharp Programming
   4. Java Programming
2. Program Testing
3. Unit Test Development
   1. C
   2. Cpp
   3. Csharp
   4. Java
4. TDD
5. Test and Deployment Automation Management

**Week-5 ( C Functional Console Programming)**

1. Programming Development
   * 1. Debugging
     2. Console Application Development
        1. C Programming
           1. C Introduction

Keywords & Identifiers

Variables & Constants

C Data Types

C Input/Output

C Operators

C Introduction Examples (homework)

* + - * 1. C Flow Control

C if..else

C for loop

C while loop

C break and continue

C switch...case

C Programming goto

Control Flow Examples (homework)

* + - * 1. C Functions

C Programming Functions

C User-defined Functions

C Function Types

C Recursion

C Storage Class

C Function Examples

* + - * 1. C Programming Arrays

C Programming Arrays

C Multi-dimensional Arrays

C Arrays & Functions

* + - * 1. C Programming Pointers

C Programming Pointers

C Pointers & Arrays

C Pointers and Functions

C Memory Allocation

Array & Pointer Examples

* + - * 1. C Programming Strings

C Programming Strings

C String Functions

C String Examples

* + - * 1. C Structure and Union

C Structure

C Struct & Pointers

C Struct & Functions

C Unions

C Struct Examples

* + - * 1. C Programming Files

C Files Input/Output

C Files Examples

* + - * 1. Additional Topics

C Enumeration

C Preprocessors

C Standard Library

C Programming Examples

**Week-6 (Cpp Functional Console Programming)**

* + - 1. Cpp Programming
         1. C++ Introduction

C++ Variables and Literals

C++ Data Types

C++ Basic I/O

C++ Type Conversion

C++ String to Int and Vice-Versa

C++ String to Float, Double and Vice-Versa

C++ Operators

C++ Comments

* + - * 1. C++ Flow Control

C++ if..else

C++ for loop

C++ do..while loop

C++ break statement

C++ continue statement

C++ switch statement

C++ goto statement

* + - * 1. C++ Functions

C++ Functions

C++ Function Types

C++ Function Overloading

C++ Default Argument

C++ Storage Class

C++ Recursion

C++ Return Reference

* + - * 1. C++ Arrays & String

C++ Arrays

Multidimensional Arrays

C++ Function and Array

C++ String

* + - * 1. C++ Structures

C++ Structures

Structure and Function

C++ Pointers to Structure

C++ Enumeration

* + - * 1. C++ Object & Class

C++ Objects and Class

C++ Constructors

C++ Objects & Function

C++ Operator Overloading

* + - * 1. C++ Pointers

C++ Pointer

C++ Pointers and Arrays

C++ Pointers and Functions

C++ Memory Management

* + - * 1. C++ Inheritance

C++ Inheritance

Inheritance Access Control

C++ Function Overriding

Multiple & Multilevel Inheritance

C++ Friend Function

C++ Virtual Function

C++ Templates

**Week-7 (Csharp Functional Console Programming)**

1. Csharp Programming
   1. Introduction
      1. C# Hello World
      2. C# Keywords & Identifiers
      3. C# Variables
      4. C# Operators
      5. C# Basic I/O
      6. C# Expressions & Statements
      7. C# Comments
   2. Flow Control
      1. C# if..else
      2. C# for loop
      3. C# while loop
      4. C# foreach loop
      5. C# switch statement
      6. C# ternary operator
   3. Exception Handling
   4. Other Topics
      1. C# Bitwise Operators
      2. C# Preprocessor Directives
      3. C# Namespaces
      4. C# Partial Class & Method

**Week-8 (Midterm)**

**Week-9 (Java Functional Console Programming)**

* + - 1. Java Programming
         1. Java Introduction

Java Hello World

Java JVM, JRE, and JDK

Java Variables

Java Data Types

Java Operators

Java Input and Output

Java Expressions & Blocks

Java Comment

* + - * 1. Java Flow Control

Java if..else

Java switch statement

Java for loop

Java for-each loop

Java while loop

Java break statement

Java continue statement

* + - * 1. Java Arrays

Java Arrays

Multidimensional Array

Java Copy Array

**Week-10 (Java Functional Console Programming)**

1. Java OOP-I
2. Java Class and Objects
3. Java Methods
4. Java Constructor
5. Java Strings
6. Java Access Modifiers
7. Java this keyword
8. Java final keyword
9. Java recursion
10. Java instanceof operator
11. Java OOP-II
12. Java Inheritance
13. Java Method Overriding
14. Java super keyword
15. Abstract Class & Method
16. Java Interfaces
17. Java Polymorphism
18. Java Encapsulation
19. Java OOP-III
20. Nested & Inner Class
21. Java Static Class
22. Java Anonymous Class
23. Java Singleton
24. Java enum class
25. Java enum constructor
26. Java enum string
27. Java reflection
28. Java Exception Handling
29. Java Exceptions
30. Java Exception Handling
31. Java try..catch
32. Java throw and throws
33. Java catch Multiple Exceptions
34. Java try-with-resources
35. Java Annotations
36. Java Annotation Types
37. Java Logging
38. Java Assertions
39. Java List
40. Java Collection Framework
41. Java Collection Interface
42. Java List Interface
43. Java ArrayList
44. Java Vector
45. Java Stack
46. Java Queue
47. Java Queue Interface
48. Java PriorityQueue Interface
49. Java Deque Interface
50. Java LinkedList
51. Java ArrayDeque
52. Java BlockingQueue Interace
53. Java ArrayBlockingQueue
54. Java LinkedBlocking Queue
55. Java Map
56. Java Map Interface
57. Java HashMap
58. Java LinkedHashMap
59. Java WeakHashMap
60. Java EnumMap
61. Java SortedMap Interface
62. Java NavigableMap Interface
63. Java TreeMap
64. Java ConcurrentMap Interface
65. Java ConcurrentHashMap
66. Java Set
67. Java Set Interface
68. Java HashSet
69. Java EnumSet
70. Java LinkedHashSet
71. Java SortedSet Interface
72. Java NavigableSet Interface
73. Java TreeSet
74. Java Algorithms
75. Java Iterator
76. Java ListIterator

**Week-11 (Java Functional Console Programming)**

1. Java I/O Streams
2. Java I/O Streams
3. Java InputStream
4. Java OutputStream
5. Java FileInputStream
6. Java FileOutputStream
7. Java ByteArrayInputStream
8. Java ByteArrayOutputStream
9. Java ObjectInputStream
10. Java ObjectOutputStream
11. Java BufferedInputStream
12. Java BufferedOutputStream
13. Java PrintStream
14. Java Reader/Writer
15. Java Reader
16. Java Writer
17. Java InputStreamReader
18. Java OutputStreamWriter
19. Java FileReader
20. Java FileWriter
21. Java BufferedReader
22. Java BufferedWriter
23. Java StringReader
24. Java StringWriter
25. Java PrintWriter
26. Additional Topics
27. Java Scanner Class
28. Java Type Casting
29. Java autoboxing and unboxing
30. Java Lambda Expression
31. Java Generics
32. Java File Class
33. Java Wrapper Class
34. Java Command Line Arguments
35. JNLP (Java Network Launch Protocol)

**Week-12 (C/Cpp GUI Programming)**

1. GUI Application Development (Windows)
   1. C (with GTK) Programming
   2. Cpp Programming

**Week-13** **(Csharp GUI Programming)**

1. GUI Application Development (Windows)
   1. Csharp Programming

**Week-14 (Csharp GUI Programming)**

1. GUI Application Development (Windows)
   1. Csharp Programming

**Week-15 (Java GUI Programming)**

1. GUI Application Development (Windows)
   1. Java Programming

**Week-16 (Final)**

**Algorithms and Programming – II**

**Week-1**

1. Course Plan and Communication
2. Grading System, Home works, and Exams.
3. Algorithms
   1. Algorithm Basics
   2. Introduction to Analysis of Algorithms
      1. Algorithm Basics
      2. Flowgorithm
      3. Pseudocode
      4. Sorting Problem
      5. Insertion Sort Analysis
      6. Algorithm Cost Calculation for Time Complexity
      7. Worst, Average, and Best Case Summary
      8. Merge Sort Analysis

**Week-2**

1. Solving Recurrences
   1. Recursion Tree
   2. Master Method
   3. Back-Substitution
2. Divide-and-Conquer Analysis
   1. Merge Sort
   2. Binary Search
   3. Merge Sort Analysis
   4. Complexity
3. Recurrence Solution

**Week-3**

1. RAM (Random Access Machine Model)
2. Asymptotic Notation
   1. Big O Notation
   2. Big Teta Notation
   3. Big Omega Notation
   4. Small o Notation
   5. Small omega Notation
3. Matrix Multiplication
   1. Traditional
   2. Recursive
   3. Strassen

**Week-4**

1. Quicksort
   1. Hoare Partitioning
   2. Lomuto Partitioning
   3. Recursive Sorting
2. Quicksort Analysis
3. Randomized Quicksort
4. Randomized Selection
   1. Recursive
   2. Medians
5. Heaps
   1. Max / Min Heap
   2. Heap Data Structure
   3. Heapify
      1. Iterative
      2. Recursive
   4. Extract-Max
   5. Build Heap
6. Heap Sort
7. Priority Queues
8. Linked Lists
9. Radix Sort
10. Counting Sort

**Week-5**

1. Convex Hull (Divide & Conquer)
2. Dynamic Programming
   1. Introduction
      1. Divide-and-Conquer (DAC) vs Dynamic Programming (DP)
      2. Fibonacci Numbers
         1. Recursive Solution
         2. Bottom-Up Solution
      3. Optimization Problems
      4. Development of a DP Algorithms
   2. Matrix-Chain Multiplication
      1. Matrix Multiplication and Row Columns Definitions
      2. Cost of Multiplication Operations (pxqxr)
      3. Counting the Number of Parenthesizations
      4. The Structure of Optimal Parenthesization
         1. Characterize the structure of an optimal solution
         2. A Recursive Solution
            1. Direct Recursion Inefficiency.
         3. Computing the optimal Cost of Matrix-Chain Multiplication
         4. Bottom-up Computation
      5. Algorithm for Computing the Optimal Costs
         1. MATRIX-CHAIN-ORDER
      6. Construction and Optimal Solution
         1. MATRIX-CHAIN-MULTIPLY
      7. Summary

**Week-6**

1. Elements of Dynamic Programming
   1. Optimal Substructure
   2. Overlapping Subproblems
2. Recursive Matrix Chain Order Memoization
   1. Top-Down Approach
   2. RMC
   3. MemoizedMatrixChain
      1. LookupC
   4. Dynamic Programming vs Memoization Summary
3. Dynamic Programming
   1. Problem-2 : Longest Common Subsequence
      1. Definitions
      2. LCS Problem
      3. Notations
      4. Optimal Substructure of LCS
         1. Proof Case-1
         2. Proof Case-2
         3. Proof Case-3
      5. A recursive solution to subproblems (inefficient)
      6. Computing the length of and LCS
         1. LCS Data Structure for DP
         2. Bottom-Up Computation
      7. Constructing and LCS
         1. PRINT-LCS
         2. Back-pointer space optimization for LCS length
4. Most Common Dynamic Programming Interview Questions
   1. Problem-1:  Longest Increasing Subsequence
      1. [https://www.geeksforgeeks.org/longest-increasing-subsequence-dp-3/](https://www.geeksforgeeks.org/longest-increasing-subsequence-dp-3/%20)
      2. [https://en.wikipedia.org/wiki/Longest\_increasing\_subsequence#:~:text=In%20computer%20science%2C%20the%20longest,not%20necessarily%20contiguous%2C%20or%20unique.](https://en.wikipedia.org/wiki/Longest_increasing_subsequence#:~:text=In%20computer%20science%2C%20the%20longest,not%20necessarily%20contiguous%2C%20or%20unique. )
      3. [https://www.youtube.com/watch?v=22s1xxRvy28&ab\_channel=StableSort](https://www.youtube.com/watch?v=22s1xxRvy28&ab_channel=StableSort%20)
   2. Problem-2: Edit Distance
      1. [https://www.geeksforgeeks.org/edit-distance-dp-5/](https://www.geeksforgeeks.org/edit-distance-dp-5/%20)
      2. [https://www.youtube.com/watch?v=tU2f2JwHmfQ&feature=youtu.be&ab\_channel=PrepForTech](https://www.youtube.com/watch?v=tU2f2JwHmfQ&feature=youtu.be&ab_channel=PrepForTech%20)
      3. Recursive
         1. [https://www.youtube.com/watch?v=8Q2IEIY2pDU&ab\_channel=BenLangmead](https://www.youtube.com/watch?v=8Q2IEIY2pDU&ab_channel=BenLangmead%20)
      4. DP
         1. [https://www.youtube.com/watch?v=0KzWq118UNI&ab\_channel=BenLangmead](https://www.youtube.com/watch?v=0KzWq118UNI&ab_channel=BenLangmead%20)
         2. [https://www.youtube.com/watch?v=eAVGRWSryGo&ab\_channel=BenLangmead](https://www.youtube.com/watch?v=eAVGRWSryGo&ab_channel=BenLangmead%20)
   3. Problem-3: Partition a set into two subsets such that the difference of subset sums is minimum
      1. [https://www.geeksforgeeks.org/partition-a-set-into-two-subsets-such-that-the-difference-of-subset-sums-is-minimum/](https://www.geeksforgeeks.org/partition-a-set-into-two-subsets-such-that-the-difference-of-subset-sums-is-minimum/%20)
   4. Problem-4: Count number of ways to cover a distance
      1. [https://www.geeksforgeeks.org/count-number-of-ways-to-cover-a-distance/](https://www.geeksforgeeks.org/count-number-of-ways-to-cover-a-distance/%20)
   5. Problem-5: Find the longest path in a matrix with given constraints
      1. [https://www.geeksforgeeks.org/find-the-longest-path-in-a-matrix-with-given-constraints/](https://www.geeksforgeeks.org/find-the-longest-path-in-a-matrix-with-given-constraints/%20)
   6. Problem-6: Subset Sum Problem
      1. [https://www.geeksforgeeks.org/subset-sum-problem-dp-25/](https://www.geeksforgeeks.org/subset-sum-problem-dp-25/%20)
   7. Problem-7: Optimal Strategy for a Game
      1. [https://www.geeksforgeeks.org/optimal-strategy-for-a-game-dp-31/](https://www.geeksforgeeks.org/optimal-strategy-for-a-game-dp-31/%20)
   8. Problem-8: 0-1 Knapsack Problem
      1. [https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/](https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/%20)
   9. Problem-9: Boolean Parenthesization Problem
      1. [https://www.geeksforgeeks.org/boolean-parenthesization-problem-dp-37/](https://www.geeksforgeeks.org/boolean-parenthesization-problem-dp-37/%20)
   10. Problem-10: Shortest Common Supersequence
       1. [https://www.geeksforgeeks.org/shortest-common-supersequence/](https://www.geeksforgeeks.org/shortest-common-supersequence/%20)
       2. [https://en.wikipedia.org/wiki/Shortest\_common\_supersequence\_problem](https://en.wikipedia.org/wiki/Shortest_common_supersequence_problem%20)
   11. Problem-11: Partition Problem
       1. [https://www.geeksforgeeks.org/partition-problem-dp-18/](https://www.geeksforgeeks.org/partition-problem-dp-18/%20)
   12. Problem-12: Cutting a Rod
       1. [https://www.geeksforgeeks.org/cutting-a-rod-dp-13/](https://www.geeksforgeeks.org/cutting-a-rod-dp-13/%20)
   13. Problem-13: Coin Change
       1. [https://www.geeksforgeeks.org/coin-change-dp-7/](https://www.geeksforgeeks.org/coin-change-dp-7/%20)
   14. Problem-14: Word Break Problem
       1. [https://www.geeksforgeeks.org/word-break-problem-dp-32/](https://www.geeksforgeeks.org/word-break-problem-dp-32/%20)
   15. Problem-15: Maximum Product Cutting
       1. [https://www.geeksforgeeks.org/maximum-product-cutting-dp-36/](https://www.geeksforgeeks.org/maximum-product-cutting-dp-36/%20)
   16. Problem-16: Dice Throw
       1. [https://www.geeksforgeeks.org/dice-throw-dp-30/](https://www.geeksforgeeks.org/dice-throw-dp-30/%20)
   17. Problem-17: Box Stacking Problem
       1. [https://www.geeksforgeeks.org/box-stacking-problem-dp-22/](https://www.geeksforgeeks.org/box-stacking-problem-dp-22/%20)
   18. Problem-18: Egg Dropping Puzzle
       1. [https://www.geeksforgeeks.org/egg-dropping-puzzle-dp-11/](https://www.geeksforgeeks.org/egg-dropping-puzzle-dp-11/%20)

**Week-7**

1. Greedy Algorithms and Dynamic Programming Differences
2. Greedy Algorithms
   1. Activity Selection Problem
   2. Knapsack Problems
      1. The 0-1 knapsack problem
      2. The fractional knapsack problem

**Week-8 (Midterm)**

**xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

**Week-9**

1. Heap Data Structure
2. Heap Sort
3. Huffman Coding

**Week-10**

1. Introduction to Graphs
2. Graphs and Representation
3. BFS (Breath-First Search)
4. DFS (Depth-First Search)
   1. in-order
   2. post-order
   3. pre-order
5. Topological Order
6. SCC (Strongly Connected Components)
7. MST
   1. Prim
   2. Kruskal

**Week-11**

1. Disjoint Sets and Kruskal Relationships
2. Single-Source Shortest Paths
   1. Bellman-Ford
   2. Dijkstra
3. Q-Learning Shortest Path
4. Max-Flow Min-Cut
   1. Ford-Fulkerson
   2. Edmond’s Karp
   3. Dinic

**Week-12**

1. Crypto++ Library Usage
2. Hashing and Encryption
   1. Integrity Control
      1. Hash Values
         1. Cryptographic Hash Functions
            1. SHA-1
            2. SHA-256
            3. SHA-512
         2. Checksums
            1. MD5
            2. CRC32
         3. Hash Algorithms
            1. SHA-1
            2. SHA-256
            3. SHA-512
            4. H-MAC

**Week-13**

1. Symmetric Encryption Algorithms
   1. AES
      1. [https://formaestudio.com/portfolio/aes-animation/](https://formaestudio.com/portfolio/aes-animation/%20)
   2. DES
      1. <http://desalgorithm.yolasite.com/>
   3. TDES
      1. [https://en.wikipedia.org/wiki/Triple\_DES](https://en.wikipedia.org/wiki/Triple_DES%20)
2. Symmetric Encryption Modes
   1. [https://en.wikipedia.org/wiki/Block\_cipher\_mode\_of\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation%20%20)
   2. ECB
   3. CBC
3. Asymmetric Encryption
   1. Key Pairs (Public-Private Key Pairs)
4. Signature Generation and Validation

**Week-14**

1. OTP Calculation
   1. Time-based
   2. Counter-based
2. File Encryption and Decryption and Integrity Control Operations

**Week-15**

1. Review

**Week-16 (Final)**

**xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

**Data Structures**

<https://www.youtube.com/c/WilliamFiset-videos/playlists>

<https://github.com/williamfiset/Algorithms>

<http://www.btechsmartclass.com/data_structures/introduction-to-algorithms.html>

https://www.programiz.com/dsa/data-structure-types

**Week-1**

1. Introduction to Data Structure
   1. Data-in-use
   2. Data-in-transit
   3. Data-at-rest
2. Performance Analysis
3. Space Complexity
4. Time Complexity
5. Data and Variables
6. Implementing Pointer and Objects
7. Linear & Non-Linear Data Structures
8. ASN.1 / BER TLV / PER TLV

**Week-2**

1. Single Linked List
2. Circular Linked List
3. Double Linked List
4. XOR Linked List
5. Skip List
6. Strand Sort
7. Arrays
   1. Array Rotations
   2. Arrangement Rearrangement
   3. Searching and Sorting
   4. Optimization Problems
8. Matrix
9. Sparse Matrix

**Week-3**

1. Stack ADT
2. Stack Using Array
3. Stack Using Linked List
4. Expressions
   1. Infix
   2. Postfix
   3. Prefix
5. Infix to Postfix Conversion
6. Postfix Expression Evaluation
7. Queue ADT
   1. First Come First Serve, FCFS, FIFO
8. Queue Datastructure Using Array
9. Queue Using Linked List
10. Circular Queue Datastructure
11. Double Ended Queue Datastructure
12. Hanoi Tower
13. Multilevel Queue (MLQ)

**Week-4**

1. Tree – Terminology
2. Tree Representations
3. Binary Tree Datastructure
   1. Construction and Conversion
   2. Checking and Printing
   3. Summation
   4. Longest Common Ancestor
4. Binary Tree Representations
5. Binary Tree Traversals
   1. In-Order
   2. Pre-Order
   3. Post-Order
6. Threaded Binary Trees
7. Max Priority Queue
8. Heap Data Structure
   1. Max-Heap
   2. Min-Heap
   3. Binary Heap
   4. Binomial Heap
   5. Fibonacci Heap
      1. Structure of Fibonacci Heaps
      2. Mergeable-heap operations
      3. Decreasing a key and deleting a node
      4. Bounding the maximum degree
   6. Leftist Heap
   7. K-ary Heap
   8. Heap Sort
   9. Huffman Coding

**Week-5**

1. Introduction to Graphs
   1. Vertex
   2. Edge
   3. Undirected Graph
   4. Directed Graph
   5. Mixed Graph
   6. End Vertices or Endpoints
   7. Origin
   8. Destination
   9. Adjacent
   10. Incident
   11. Outgoing Edge
   12. Incoming Edge
   13. Degree
   14. Indegree
   15. Outdegree
   16. Parallel edges or Multiple edges
   17. Self-loop
   18. Simple Graph
   19. Path
2. Graph Representations
   1. Adjacency Matrix
   2. Incidence Matrix
   3. Adjacency List
3. Graph Traversal
   1. Depth-First Search (DFS)
      1. Iterative Deepening Search(IDS) or Iterative Deepening Depth First Search(IDDFS)
   2. Breadth-First Search (BFS)
   3. Depth-limited Search
   4. Uniform Cost Search
   5. Bidirectional Search
   6. Water Jug Problem

**Week-6**

1. Graph Topological Sorting
2. Graph MST
3. Graph Backtracking
   1. Tug of War
   2. n-Queen's Problem
   3. m Coloring Problem
   4. Euler & Hamiltonian Path
4. Graph Sortest Paths
5. Graph Connectivity
6. Graph Max Flow
7. Graph Isomorphism
   1. <https://github.com/Mith13/Graphs-isomorphism>
8. Graph canonization
9. Graph Cuts
   1. Min Cut
   2. Max Cut
10. Alpha-Beta Pruning
11. Hasse Diagrams
12. Petri Nets
13. Bipartite Graphs
14. Cycle Detection
    1. Brent’s Algorithm
    2. Hare and Tortoise Algorithm
15. Bayesian Network

**Week-7**

1. Linear Search
2. Binary Search
   1. Interpolation Search
3. Fibonacci Search
4. Hashing and Hash Tables
   1. Direct-Address Tables
   2. Hash Tables
   3. Hash Functions
   4. Open Adressing
   5. Perfect Hashing

**Week-8 (Midterm)**

**xxxxxxxxxxxxxxxxxxxxxxxxxxx**

**Week-9**

1. Sortings
   1. Insertion Sort
   2. Selection Sort
   3. Radix Sort
   4. Quick Sort
   5. Heap Sort
   6. Permutation Sort
   7. Gnome Sort
   8. Comb Sort
   9. Flash Sort
   10. Stooge Sort
   11. Bees Algorithm
   12. Lucky Sort
   13. Indirect Sort (Pointer Sort)
   14. External Sort (Segmented Sort)
   15. Shaker Sort / Bidirectional Bubble Sort
   16. Shell Sort
   17. Comparison of Sorting Methods

**Week-10**

1. Trees
   1. Binary Search Tree
      1. Search and Insertion
      2. Delete
      3. BST over Hash Table
      4. Construction and Conversions
      5. Check Smallest/Largest Element
      6. Red Black Tree and Threaded Binary Tree
   2. AVL Trees
   3. B Trees
      1. Defitinion of B Trees
      2. Basic operations on B tree
      3. Deleting a key from a B tree
   4. 2 3 4 Trees
   5. 2 3 Trees
   6. B+ Trees
   7. R Trees
   8. Red - Black Tree Datastructure
   9. Splay Tree Datastructure
   10. Augmenting Data Structures
       1. Dynamic order statistics
       2. How to augment a data structure
       3. Interval trees
   11. van Emde Boas Trees
       1. Preliminary approaches
       2. A recursive structure
       3. The van Emde Boas tree
   12. Binomial Trees
   13. Comparison of Search Trees
   14. Minimax Tree

**Week-11**

1. Strings
   1. Longest common subsequence problem
      1. Longest increasing subsequence
      2. Hunt–Szymanski algorithm (Hunt Macllory)
      3. Levenshtein distance
      4. Wagner–Fischer algorithm
   2. String Alignment
      1. Needleman Wunsch
      2. Smith Waterman
      3. Hunt Macllory
   3. String Tokenizer
   4. String Comparison

**Week-12**

1. Strings
   1. Reverse Factor Algorithm (String Search)
      1. Knuth-Morris-Pratt Algorithm
      2. Horspool Algorithm
      3. Boyer-Moore Algorithm
      4. Brute-Force / Linear Text Search
      5. DFA Text Search
2. Tries
   1. Patricia Tree (Radix Tree)
3. Data Structures for Disjoint Sets
   1. Disjoint-set operations
   2. Linked-list representation of disjoint sets
   3. Disjoint-set forests
   4. Analysis of union by rank with path compression

**Week-13**

1. File Organization
   1. Sequential File Organization
      1. Binary Search
      2. Interpolation Search
      3. Self-Organizing Sequential Search
   2. Direct File Organization
      1. Locating Information
      2. Hashing Functions (MD5, HAVAL, SHA1 etc.)
         1. Key mod N
         2. Key mod P
         3. Truncation
         4. Folding
         5. Squaring
         6. Radix Conversion
         7. Polynomial Hashing
         8. Alphabetic Keys
         9. Collisions
      3. Collision Resolution
         1. Collision resolution with links
         2. Collision resolution without links
            1. Static positioning of records
            2. Dynamic positioning of records
         3. Collision resolution with pseudolinks
      4. Coalesced Hashing
         1. EISCH
         2. LISCH
         3. BEISCH
         4. BLISCH
         5. REISCH
         6. RLISCH
         7. EICH
         8. LICH
      5. Progressive Overflow
         1. Linear Probing
         2. Quadratic Probing
      6. Double Hashing
      7. Use of Buckets
      8. Linear Quotient
      9. Brent’s Method
      10. Binary Tree
      11. Computed Chaining Insertion(CCI)
      12. Comparison of Collision Resolution Methods
      13. Perfect Hashing
      14. SimHash

**Week-14**

1. Indexed Sequential File Organization
2. Bits of Information
3. Secondary Key Retrieval
4. Multilist File Organization
5. Inverted Files
6. Partial Match Retrieval with Signature Trees
7. Partial Match Retrieval with Page Signatures
8. Bits and Hashing
9. Signature Hashing
10. Bloom Filters
11. Classification Hashing
12. Check Hashing
13. Binary Tree Structures
14. Binary Search Trees
15. AVL Trees
16. Internal Path Reduction Trees
17. B-Trees and Derivatives
18. B-Trees
19. B#-Trees
20. B+ -Trees

**Week-15**

1. Hashing Techniques for Expandable Files
2. Extendible Hashing
3. Dynamic Hashing
4. Linear Hashing
5. Other Tree Structures
6. Tries
7. Approximate String Matching
8. Trie Hashing
9. PATRICIA Trees
10. Digital Search Trees
11. Seconday Key Retrieval (2)
12. K-d trees
13. Grid Files
14. File Sorting
15. Insertion Sort
16. Quicksort
17. Heapsort
18. External Sorting
19. Sorting by Merging
20. Disk Sort

**Week-16 (Final)**

**xxxxxxxxxxxxxxxxxxxxxxxxxxx**