

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-algoritmaları-search-algorithms/>

<https://bilgisayarkavramlari.com/2008/08/09/siralama-algoritmaları-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.learn-c.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)

Compiler, 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.csharpnadir.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/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://en.wikipedia.org/wiki/Longest_common_subsequence_problem

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=B2AJoSzf4M&ab_channel=StableSort

Greedy Algorithms

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

0-1 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=CUAoIXf8u-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/Cyclic_redundancy_check

<https://en.wikipedia.org/wiki/MD5>

<https://en.wikipedia.org/wiki/SHA-1>

<https://en.wikipedia.org/wiki/SHA-2>

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
 - a. Operating System Basics
 - i. Types of Operating Systems
 - ii. Console commands
 - iii. System folders
 - iv. System parameters
 - v. Storage management
 - b. Computer Network Basics
 - i. Network connections
 - ii. Network related console commands (ipconfig, ipconfig /renew, ipconfig /release, hostname, netstat -a, nslookup)
 - iii. IP, Port, DNS settings, NAT etc.
 - iv. Remote connections (FTP, SSH, RDP, XRDP)
 - v. Putty, MobaXterm
 - c. Numerical System Basics
 - i. Binary system
 - ii. Hexadecimal system
 - d. Character Sets

Week-2 (Introduction to Algorithms and Development Environments)

1. Algorithm Basics
2. Introduction to Analysis of Algorithms
 - a. Algorithm Basics
 - b. Flowchart
 - c. Pseudocode
3. Programming Topics
 - a. Programming Environment Setup and Configuration
 - i. C / Cpp
 1. DevCpp
 2. Code Blocks
 3. GCC/G++ Compiler (Mingw) / Clang-cl (LLVM)
 4. vscode
 5. Visual Studio Community Edition
 - a. Visual Studio x64 x86 Configurations and Features
 - b. Project Types
 6. Notepad++
 7. Vi/Vim
 8. Eclipse
 - a. Simple Java Project Generation
 - b. Jar Export as Library or Executable

c. Maven Project Generation

d. Junit Test Case Generation and Testing

9. Netbeans

10. Turbo C

11. Turbo C++

ii. Java

1. JDK, JRE Setup

2. System Environments and Paths

3. Netbeans

4. Eclipse

5. IntelliJ Idea (jetbrains)

6. vscode

7. Notepad++

iii. C#

1. Visual Studio Community Edition

2. Notepad++

b. Programming Environment Setup and Configuration

i. Notepad++ (Notepad for Source Code)

ii. HxD (Hex Editor)

iii. Marktext (Markdown Syntax Editor)

iv. Cygwin (Linux environment for Windows)

v. Dependency Walker (32-bit or 64-bit Windows module dependency checker)

vi. Doxygen (Code Documentation)

vii. Sonarlint (Code Quality and Code Security Extension)

viii. Codepen.io (online code sharing)

ix. Codeshare.io (real time code sharing)

x. Codebeautify.org (online data conversion tools)

xi. AsciiFlow.com (ASCII drawing tool)

xii. Freemind (opensource mindmap application)

xiii. Wireflow (user flow designer)

xiv. PlantUML (software designer)

xv. Drawio (drawing tool)

xvi. Putty (Remote Connection)

xvii. MobaXterm (Remote Connection)

xviii. Teamviewer (Remote Connection)

xix. Paletton.com (Color Chooser)

xx. Understand (Static Code Analysis)

xxi. JD Project (Java Decompiler)

xxii. Cutter (Multi-Platform Reverse Engineering Tool)

xxiii. IDA Pro / Freeware (Native Reverse Engineering Tool)

xxiv. Travis-CI

1. Travis.yml

xxv. Jenkins

xxvi. Valgrind

xxvii. Docker

1. Dockerfile

2. DockerHub

3. Docker Compose Yaml

- 4. Dockerrun.aws.json (AWS)
 - xxviii. Nuget Packages
 - xxix. Extras
 - 1. vim/vim-wim32-installer (windows vim installer)
 - xxx. SCV Cryptomanager
 - xxxi. Addario CryptoBench
 - xxxii. Raymond's MD5 & SHA Checksum Utility
 - xxxiii. SlavaSoft HashCalc
 - xxxiv. Portable PGP
- c. Online Programming Envoriments
 - i. Hackerrank
 - ii. CS50 Sandbox
 - iii. Programiz C Online Compiler

Week-3 (Introduction to Source Code Version Management)

- 1. Programming Source Code Sharing and Version Management
 - a. Introduction to Source Code Management Systems
 - b. Features of Source Code Management
 - c. Why Do We Need Source Code Management?
 - d. Types of Version Control Systems
 - i. Centralized
 - ii. Distributed
 - e. Git Usage
 - i. Installation of Git
 - 1. Git
 - 2. Git-Extension
 - 3. Gig (git ignore creator)
 - ii. Configuration of Git
 - iii. Github Platform Usage
 - iv. Create a New Repository
 - v. Checkout a Repository
 - vi. Add & Commit (Write Good Commits)
 - vii. Pushing Changes
 - 1. Update Local Repo Before Sending
 - 2. Send Changes to Remote Repo
 - viii. Branching
 - ix. Update & Merge
 - 1. Fast-forward (-ff) Merging
 - 2. No-fast-forward (--no-ff) Merging
 - 3. Merge Conflicts
 - x. Rebasing
 - xi. Replace Local Changes / Resetting
 - 1. Soft reset
 - 2. Hard reset
 - 3. Reverting
 - xii. Cherry-picking
 - xiii. Fetching
 - xiv. Reflog

- xv. Tagging
- xvi. Log
- xvii. Gource
- f. Maven Usage
- g. TFS Usage
- h. SVN Usage

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

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

Week-5 (C Functional Console Programming)

- 1. Programming Development
 - i. Debugging
 - ii. Console Application Development
 - 1. C Programming
 - a. C Introduction
 - i. Keywords & Identifiers
 - ii. Variables & Constants
 - iii. C Data Types
 - iv. C Input/Output
 - v. C Operators
 - vi. C Introduction Examples (homework)
 - b. C Flow Control
 - i. C if..else
 - ii. C for loop
 - iii. C while loop
 - iv. C break and continue
 - v. C switch...case
 - vi. C Programming goto
 - vii. Control Flow Examples (homework)
 - c. C Functions
 - i. C Programming Functions
 - ii. C User-defined Functions
 - iii. C Function Types
 - iv. C Recursion
 - v. C Storage Class

- vi. C Function Examples
- d. C Programming Arrays
 - i. C Programming Arrays
 - ii. C Multi-dimensional Arrays
 - iii. C Arrays & Functions
- e. C Programming Pointers
 - i. C Programming Pointers
 - ii. C Pointers & Arrays
 - iii. C Pointers and Functions
 - iv. C Memory Allocation
 - v. Array & Pointer Examples
- f. C Programming Strings
 - i. C Programming Strings
 - ii. C String Functions
 - iii. C String Examples
- g. C Structure and Union
 - i. C Structure
 - ii. C Struct & Pointers
 - iii. C Struct & Functions
 - iv. C Unions
 - v. C Struct Examples
- h. C Programming Files
 - i. C Files Input/Output
 - ii. C Files Examples
- i. Additional Topics
 - i. C Enumeration
 - ii. C Preprocessors
 - iii. C Standard Library
 - iv. C Programming Examples

Week-6 (Cpp Functional Console Programming)

1. Cpp Programming
 - a. C++ Introduction
 - i. C++ Variables and Literals
 - ii. C++ Data Types
 - iii. C++ Basic I/O
 - iv. C++ Type Conversion
 1. C++ String to Int and Vice-Versa
 2. C++ String to Float, Double and Vice-Versa
 - v. C++ Operators
 - vi. C++ Comments
 - b. C++ Flow Control
 - i. C++ if..else
 - ii. C++ for loop
 - iii. C++ do..while loop
 - iv. C++ break statement
 - v. C++ continue statement
 - vi. C++ switch statement

- vii. C++ goto statement
- c. C++ Functions
 - i. C++ Functions
 - ii. C++ Function Types
 - iii. C++ Function Overloading
 - iv. C++ Default Argument
 - v. C++ Storage Class
 - vi. C++ Recursion
 - vii. C++ Return Reference
- d. C++ Arrays & String
 - i. C++ Arrays
 - ii. Multidimensional Arrays
 - iii. C++ Function and Array
 - iv. C++ String
- e. C++ Structures
 - i. C++ Structures
 - ii. Structure and Function
 - iii. C++ Pointers to Structure
 - iv. C++ Enumeration
- f. C++ Object & Class
 - i. C++ Objects and Class
 - ii. C++ Constructors
 - iii. C++ Objects & Function
 - iv. C++ Operator Overloading
- g. C++ Pointers
 - i. C++ Pointer
 - ii. C++ Pointers and Arrays
 - iii. C++ Pointers and Functions
 - iv. C++ Memory Management
- h. C++ Inheritance
 - i. C++ Inheritance
 - ii. Inheritance Access Control
 - iii. C++ Function Overriding
 - iv. Multiple & Multilevel Inheritance
 - v. C++ Friend Function
 - vi. C++ Virtual Function
 - vii. C++ Templates

Week-7 (Csharp Functional Console Programming)

- 1. Csharp Programming
 - a. Introduction
 - i. C# Hello World
 - ii. C# Keywords & Identifiers
 - iii. C# Variables
 - iv. C# Operators
 - v. C# Basic I/O
 - vi. C# Expressions & Statements
 - vii. C# Comments

- b. Flow Control
 - i. C# if..else
 - ii. C# for loop
 - iii. C# while loop
 - iv. C# foreach loop
 - v. C# switch statement
 - vi. C# ternary operator
- c. Exception Handling
- d. Other Topics
 - i. C# Bitwise Operators
 - ii. C# Preprocessor Directives
 - iii. C# Namespaces
 - iv. C# Partial Class & Method

Week-8 (Midterm)

Week-9 (Java Functional Console Programming)

- 1. Java Programming
 - a. Java Introduction
 - i. Java Hello World
 - ii. Java JVM, JRE, and JDK
 - iii. Java Variables
 - iv. Java Data Types
 - v. Java Operators
 - vi. Java Input and Output
 - vii. Java Expressions & Blocks
 - viii. Java Comment
 - b. Java Flow Control
 - i. Java if..else
 - ii. Java switch statement
 - iii. Java for loop
 - iv. Java for-each loop
 - v. Java while loop
 - vi. Java break statement
 - vii. Java continue statement
 - c. Java Arrays
 - i. Java Arrays
 - ii. Multidimensional Array
 - iii. Java Copy Array

Week-10 (Java Functional Console Programming)

- a. Java OOP-I
 - i. Java Class and Objects
 - ii. Java Methods
 - iii. Java Constructor
 - iv. Java Strings
 - v. Java Access Modifiers
 - vi. Java this keyword

- vii. Java final keyword
 - viii. Java recursion
 - ix. Java instanceof operator
- b. Java OOP-II
 - i. Java Inheritance
 - ii. Java Method Overriding
 - iii. Java super keyword
 - iv. Abstract Class & Method
 - v. Java Interfaces
 - vi. Java Polymorphism
 - vii. Java Encapsulation
- c. Java OOP-III
 - i. Nested & Inner Class
 - ii. Java Static Class
 - iii. Java Anonymous Class
 - iv. Java Singleton
 - v. Java enum class
 - vi. Java enum constructor
 - vii. Java enum string
 - viii. Java reflection
- d. Java Exception Handling
 - i. Java Exceptions
 - ii. Java Exception Handling
 - iii. Java try..catch
 - iv. Java throw and throws
 - v. Java catch Multiple Exceptions
 - vi. Java try-with-resources
 - vii. Java Annotations
 - viii. Java Annotation Types
 - ix. Java Logging
 - x. Java Assertions
- e. Java List
 - i. Java Collection Framework
 - ii. Java Collection Interface
 - iii. Java List Interface
 - iv. Java ArrayList
 - v. Java Vector
 - vi. Java Stack
- f. Java Queue
 - i. Java Queue Interface
 - ii. Java PriorityQueue Interface
 - iii. Java Deque Interface
 - iv. Java LinkedList
 - v. Java ArrayDeque
 - vi. Java BlockingQueue Interface
 - vii. Java ArrayBlockingQueue
 - viii. Java LinkedBlocking Queue
- g. Java Map

- i. Java Map Interface
 - ii. Java HashMap
 - iii. Java LinkedHashMap
 - iv. Java WeakHashMap
 - v. Java EnumMap
 - vi. Java SortedMap Interface
 - vii. Java NavigableMap Interface
 - viii. Java TreeMap
 - ix. Java ConcurrentMap Interface
 - x. Java ConcurrentHashMap
- h. Java Set
 - i. Java Set Interface
 - ii. Java HashSet
 - iii. Java EnumSet
 - iv. Java LinkedHashSet
 - v. Java SortedSet Interface
 - vi. Java NavigableSet Interface
 - vii. Java TreeSet
 - viii. Java Algorithms
 - ix. Java Iterator
 - x. Java ListIterator

Week-11 (Java Functional Console Programming)

- i. Java I/O Streams
 - i. Java I/O Streams
 - ii. Java InputStream
 - iii. Java OutputStream
 - iv. Java FileInputStream
 - v. Java FileOutputStream
 - vi. Java ByteArrayInputStream
 - vii. Java ByteArrayOutputStream
 - viii. Java ObjectInputStream
 - ix. Java ObjectOutputStream
 - x. Java BufferedInputStream
 - xi. Java BufferedOutputStream
 - xii. Java PrintStream
- j. Java Reader/Writer
 - i. Java Reader
 - ii. Java Writer
 - iii. Java InputStreamReader
 - iv. Java OutputStreamWriter
 - v. Java FileReader
 - vi. Java FileWriter
 - vii. Java BufferedReader
 - viii. Java BufferedWriter
 - ix. Java StringReader
 - x. Java StringWriter
 - xi. Java PrintWriter

- k. Additional Topics
 - i. Java Scanner Class
 - ii. Java Type Casting
 - iii. Java autoboxing and unboxing
 - iv. Java Lambda Expression
 - v. Java Generics
 - vi. Java File Class
 - vii. Java Wrapper Class
 - viii. Java Command Line Arguments
 - ix. JNLP (Java Network Launch Protocol)

Week-12 (C/Cpp GUI Programming)

- 6. GUI Application Development (Windows)
 - a. C (with GTK) Programming
 - b. Cpp Programming

Week-13 (Csharp GUI Programming)

- 7. GUI Application Development (Windows)
 - a. Csharp Programming

Week-14 (Csharp GUI Programming)

- 8. GUI Application Development (Windows)
 - a. Csharp Programming

Week-15 (Java GUI Programming)

- 9. GUI Application Development (Windows)
 - a. 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
 - a. Algorithm Basics
 - b. Introduction to Analysis of Algorithms
 - i. Algorithm Basics
 - ii. Flowgorithm
 - iii. Pseudocode
 - iv. Sorting Problem
 - v. Insertion Sort Analysis
 - vi. Algorithm Cost Calculation for Time Complexity
 - vii. Worst, Average, and Best Case Summary
 - viii. Merge Sort Analysis

Week-2

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

Week-3

1. RAM (Random Access Machine Model)
2. Asymptotic Notation
 - a. Big O Notation
 - b. Big Teta Notation
 - c. Big Omega Notation
 - d. Small o Notation
 - e. Small omega Notation
3. Matrix Multiplication
 - a. Traditional
 - b. Recursive
 - c. Strassen

Week-4

1. Quicksort
 - a. Hoare Partitioning
 - b. Lomuto Partitioning

- c. Recursive Sorting
- 2. Quicksort Analysis
- 3. Randomized Quicksort
- 4. Randomized Selection
 - a. Recursive
 - b. Medians
- 5. Heaps
 - a. Max / Min Heap
 - b. Heap Data Structure
 - c. Heapify
 - i. Iterative
 - ii. Recursive
 - d. Extract-Max
 - e. 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
 - a. Introduction
 - i. Divide-and-Conquer (DAC) vs Dynamic Programming (DP)
 - ii. Fibonacci Numbers
 - 1. Recursive Solution
 - 2. Bottom-Up Solution
 - iii. Optimization Problems
 - iv. Development of a DP Algorithms
 - b. Matrix-Chain Multiplication
 - i. Matrix Multiplication and Row Columns Definitions
 - ii. Cost of Multiplication Operations (pxqxr)
 - iii. Counting the Number of Parenthesizations
 - iv. The Structure of Optimal Parenthesization
 - 1. Characterize the structure of an optimal solution
 - 2. A Recursive Solution
 - a. Direct Recursion Inefficiency.
 - 3. Computing the optimal Cost of Matrix-Chain Multiplication
 - 4. Bottom-up Computation
 - v. Algorithm for Computing the Optimal Costs
 - 1. MATRIX-CHAIN-ORDER
 - vi. Construction and Optimal Solution
 - 1. MATRIX-CHAIN-MULTIPLY
 - vii. Summary

Week-6

- 1. Elements of Dynamic Programming

- a. Optimal Substructure
 - b. Overlapping Subproblems
- 2. Recursive Matrix Chain Order Memoization
 - a. Top-Down Approach
 - b. RMC
 - c. MemoizedMatrixChain
 - i. LookupC
 - d. Dynamic Programming vs Memoization Summary
- 3. Dynamic Programming
 - a. Problem-2 : Longest Common Subsequence
 - i. Definitions
 - ii. LCS Problem
 - iii. Notations
 - iv. Optimal Substructure of LCS
 - 1. Proof Case-1
 - 2. Proof Case-2
 - 3. Proof Case-3
 - v. A recursive solution to subproblems (inefficient)
 - vi. Computing the length of and LCS
 - 1. LCS Data Structure for DP
 - 2. Bottom-Up Computation
 - vii. Constructing and LCS
 - 1. PRINT-LCS
 - 2. Back-pointer space optimization for LCS length
- 4. Most Common Dynamic Programming Interview Questions
 - a. Problem-1: Longest Increasing Subsequence
 - i. <https://www.geeksforgeeks.org/longest-increasing-subsequence-dp-3/>
 - ii. https://en.wikipedia.org/wiki/Longest_increasing_subsequence#:~:text=In%20computer%20science%2C%20the%20longest,not%20necessarily%20contiguous%2C%20or%20unique.
 - iii. https://www.youtube.com/watch?v=22s1xxRvy28&ab_channel=StableSort
 - b. Problem-2: Edit Distance
 - i. <https://www.geeksforgeeks.org/edit-distance-dp-5/>
 - ii. https://www.youtube.com/watch?v=tU2f2JwHmfQ&feature=youtu.be&ab_channel=PrepForTech
 - iii. Recursive
 - 1. https://www.youtube.com/watch?v=8Q2IEIY2pDU&ab_channel=BenLangmead
 - iv. DP
 - 1. https://www.youtube.com/watch?v=0KzWq118UNI&ab_channel=BenLangmead
 - 2. https://www.youtube.com/watch?v=eAVGRWSryGo&ab_channel=BenLangmead
 - c. Problem-3: Partition a set into two subsets such that the difference of subset sums is minimum
 - i. <https://www.geeksforgeeks.org/partition-a-set-into-two-subsets-such-that-the-difference-of-subset-sums-is-minimum/>
 - d. Problem-4: Count number of ways to cover a distance

- i. <https://www.geeksforgeeks.org/count-number-of-ways-to-cover-a-distance/>
- e. Problem-5: Find the longest path in a matrix with given constraints
 - i. <https://www.geeksforgeeks.org/find-the-longest-path-in-a-matrix-with-given-constraints/>
- f. Problem-6: Subset Sum Problem
 - i. <https://www.geeksforgeeks.org/subset-sum-problem-dp-25/>
- g. Problem-7: Optimal Strategy for a Game
 - i. <https://www.geeksforgeeks.org/optimal-strategy-for-a-game-dp-31/>
- h. Problem-8: 0-1 Knapsack Problem
 - i. <https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/>
- i. Problem-9: Boolean Parenthesization Problem
 - i. <https://www.geeksforgeeks.org/boolean-parenthesization-problem-dp-37/>
- j. Problem-10: Shortest Common Supersequence
 - i. <https://www.geeksforgeeks.org/shortest-common-supersequence/>
 - ii. https://en.wikipedia.org/wiki/Shortest_common_supersequence_problem
- k. Problem-11: Partition Problem
 - i. <https://www.geeksforgeeks.org/partition-problem-dp-18/>
- l. Problem-12: Cutting a Rod
 - i. <https://www.geeksforgeeks.org/cutting-a-rod-dp-13/>
- m. Problem-13: Coin Change
 - i. <https://www.geeksforgeeks.org/coin-change-dp-7/>
- n. Problem-14: Word Break Problem
 - i. <https://www.geeksforgeeks.org/word-break-problem-dp-32/>
- o. Problem-15: Maximum Product Cutting
 - i. <https://www.geeksforgeeks.org/maximum-product-cutting-dp-36/>
- p. Problem-16: Dice Throw
 - i. <https://www.geeksforgeeks.org/dice-throw-dp-30/>
- q. Problem-17: Box Stacking Problem
 - i. <https://www.geeksforgeeks.org/box-stacking-problem-dp-22/>
- r. Problem-18: Egg Dropping Puzzle
 - i. <https://www.geeksforgeeks.org/egg-dropping-puzzle-dp-11/>

Week-7

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

Week-8 (Midterm)

xx

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)
 - a. in-order
 - b. post-order
 - c. pre-order
5. Topological Order
6. SCC (Strongly Connected Components)
7. MST
 - a. Prim
 - b. Kruskal

Week-11

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

Week-12

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

Week-13

1. Symmetric Encryption Algorithms

[illegible]

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
 - a. Data-in-use
 - b. Data-in-transit
 - c. 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
 - a. Array Rotations
 - b. Arrangement Rearrangement
 - c. Searching and Sorting
 - d. Optimization Problems
8. Matrix
9. Sparse Matrix

Week-3

1. Stack ADT
2. Stack Using Array
3. Stack Using Linked List
4. Expressions
 - a. Infix
 - b. Postfix
 - c. Prefix
5. Infix to Postfix Conversion
6. Postfix Expression Evaluation

7. Queue ADT
 - a. 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
 - a. Construction and Conversion
 - b. Checking and Printing
 - c. Summation
 - d. Longest Common Ancestor
4. Binary Tree Representations
5. Binary Tree Traversals
 - a. In-Order
 - b. Pre-Order
 - c. Post-Order
6. Threaded Binary Trees
7. Max Priority Queue
8. Heap Data Structure
 - a. Max-Heap
 - b. Min-Heap
 - c. Binary Heap
 - d. Binomial Heap
 - e. Fibonacci Heap
 - i. Structure of Fibonacci Heaps
 - ii. Mergeable-heap operations
 - iii. Decreasing a key and deleting a node
 - iv. Bounding the maximum degree
 - f. Leftist Heap
 - g. K-ary Heap
 - h. Heap Sort
 - i. Huffman Coding

Week-5

1. Introduction to Graphs
 - a. Vertex
 - b. Edge
 - c. Undirected Graph
 - d. Directed Graph
 - e. Mixed Graph
 - f. End Vertices or Endpoints
 - g. Origin

- h. Destination
 - i. Adjacent
 - j. Incident
 - k. Outgoing Edge
 - l. Incoming Edge
 - m. Degree
 - n. Indegree
 - o. Outdegree
 - p. Parallel edges or Multiple edges
 - q. Self-loop
 - r. Simple Graph
 - s. Path
2. Graph Representations
 - a. Adjacency Matrix
 - b. Incidence Matrix
 - c. Adjacency List
 3. Graph Traversal
 - a. Depth-First Search (DFS)
 - i. Iterative Deepening Search(IDS) or Iterative Deepening Depth First Search(IDDFS)
 - b. Breadth-First Search (BFS)
 - c. Depth-limited Search
 - d. Uniform Cost Search
 - e. Bidirectional Search
 - f. Water Jug Problem

Week-6

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

- a. Brent's Algorithm
 - b. Hare and Tortoise Algorithm
15. Bayesian Network

Week-7

- 1. Linear Search
- 2. Binary Search
 - a. Interpolation Search
- 3. Fibonacci Search
- 4. Hashing and Hash Tables
 - a. Direct-Address Tables
 - b. Hash Tables
 - c. Hash Functions
 - d. Open Addressing
 - e. Perfect Hashing

Week-8 (Midterm)

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Week-9

- 5. Sortings
 - a. Insertion Sort
 - b. Selection Sort
 - c. Radix Sort
 - d. Quick Sort
 - e. Heap Sort
 - f. Permutation Sort
 - g. Gnome Sort
 - h. Comb Sort
 - i. Flash Sort
 - j. Stooge Sort
 - k. Bees Algorithm
 - l. Lucky Sort
 - m. Indirect Sort (Pointer Sort)
 - n. External Sort (Segmented Sort)
 - o. Shaker Sort / Bidirectional Bubble Sort
 - p. Shell Sort
 - q. Comparison of Sorting Methods

Week-10

- 1. Trees
 - a. Binary Search Tree
 - i. Search and Insertion
 - ii. Delete
 - iii. BST over Hash Table
 - iv. Construction and Conversions
 - v. Check Smallest/Largest Element

- vi. Red Black Tree and Threaded Binary Tree
- b. AVL Trees
- c. B Trees
 - i. Definition of B Trees
 - ii. Basic operations on B tree
 - iii. Deleting a key from a B tree
- d. 2 3 4 Trees
- e. 2 3 Trees
- f. B+ Trees
- g. R Trees
- h. Red - Black Tree Datastructure
- i. Splay Tree Datastructure
- j. Augmenting Data Structures
 - i. Dynamic order statistics
 - ii. How to augment a data structure
 - iii. Interval trees
- k. van Emde Boas Trees
 - i. Preliminary approaches
 - ii. A recursive structure
 - iii. The van Emde Boas tree
- l. Binomial Trees
- m. Comparison of Search Trees
- n. Minimax Tree

Week-11

- 1. Strings
 - a. Longest common subsequence problem
 - i. Longest increasing subsequence
 - ii. Hunt–Szymanski algorithm (Hunt Macclory)
 - iii. Levenshtein distance
 - iv. Wagner–Fischer algorithm
 - b. String Alignment
 - i. Needleman Wunsch
 - ii. Smith Waterman
 - iii. Hunt Macclory
 - c. String Tokenizer
 - d. String Comparison

Week-12

- 2. Strings
 - a. Reverse Factor Algorithm (String Search)
 - i. Knuth-Morris-Pratt Algorithm
 - ii. Horspool Algorithm
 - iii. Boyer-Moore Algorithm
 - iv. Brute-Force / Linear Text Search
 - v. DFA Text Search
- 1. Tries
 - a. Patricia Tree (Radix Tree)

2. Data Structures for Disjoint Sets
 - a. Disjoint-set operations
 - b. Linked-list representation of disjoint sets
 - c. Disjoint-set forests
 - d. Analysis of union by rank with path compression

Week-13

1. File Organization

- a. Sequential File Organization
 - i. Binary Search
 - ii. Interpolation Search
 - iii. Self-Organizing Sequential Search
- b. Direct File Organization
 - i. Locating Information
 - ii. 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
 - iii. Collision Resolution
 1. Collision resolution with links
 2. Collision resolution without links
 - a. Static positioning of records
 - b. Dynamic positioning of records
 3. Collision resolution with pseudolinks
 - iv. Coalesced Hashing
 1. EISCH
 2. LISCH
 3. BEISCH
 4. BLISCH
 5. REISCH
 6. RLISCH
 7. EICH
 8. LICH
 - v. Progressive Overflow
 1. Linear Probing
 2. Quadratic Probing
 - vi. Double Hashing
 - vii. Use of Buckets
 - viii. Linear Quotient
 - ix. Brent's Method
 - x. Binary Tree
 - xi. Computed Chaining Insertion(CCI)

- xii. Comparison of Collision Resolution Methods
- xiii. Perfect Hashing
- xiv. SimHash

Week-14

- a. Indexed Sequential File Organization
- b. Bits of Information
- c. Secondary Key Retrieval
 - i. Multilist File Organization
 - ii. Inverted Files
 - iii. Partial Match Retrieval with Signature Trees
 - iv. Partial Match Retrieval with Page Signatures
- d. Bits and Hashing
 - v. Signature Hashing
 - vi. Bloom Filters
 - vii. Classification Hashing
 - viii. Check Hashing
- e. Binary Tree Structures
 - ix. Binary Search Trees
 - x. AVL Trees
 - xi. Internal Path Reduction Trees
- f. B-Trees and Derivatives
 - xii. B-Trees
 - xiii. B#-Trees
 - xiv. B+ -Trees

Week-15

- g. Hashing Techniques for Expandable Files
 - i. Extendible Hashing
 - ii. Dynamic Hashing
 - iii. Linear Hashing
- h. Other Tree Structures
 - iv. Tries
 - v. Approximate String Matching
 - vi. Trie Hashing
 - vii. PATRICIA Trees
 - viii. Digital Search Trees
- i. Secondary Key Retrieval (2)
 - ix. K-d trees
 - x. Grid Files
- j. File Sorting
 - xi. Insertion Sort
 - xii. Quicksort
 - xiii. Heapsort
 - xiv. External Sorting
 - xv. Sorting by Merging
 - xvi. Disk Sort

Week-16 (Final)

XXXXXXXXXXXXXXXXXXXXXXXXXXXX