Programming Languages, Object Oriented, Data Structures and Algorithms Training

Training Links

github.com/cs-MohamedAyman/Programming-Languages-and-Object-Oriented-Training

github.com/cs-MohamedAyman/Data-Structures-and-Algorithms-Training

Lecture Agenda

we will discuss in this presentation the following points

1- Why this Training is Important

2- Training Content and Educational Projects

3- How to Practice on Online Competitions

4- Technical Tutorials and Reference Textbooks

5- Online Learning Platforms Courses

6- Environment Setup and Installations

Let's
STARTUP

Lecture Agenda

1- Why this Training is Important

2- Training Content and Educational Projects

3- How to Practice on Online Competitions

4- Technical Tutorials and Reference Textbooks

5- Online Learning Platforms Courses

6- Environment Setup and Installations

Python & CPP Features

Python & CPP Features

Easy to Learn and Code

- Python is one of the most user-friendly programming languages. One can easily learn the basics of Python and become familiar with its syntax and be able to write basic programs in a few days. However, learning the advanced concepts and mastering python may take you some time.
- The syntax of Python is very easy. It generally comprises words from the English language. While writing the code it generally feels like giving the instructions to a child. Apart from this, unlike other programming languages (like C, C++, Java, etc., there is no need to take care of any opening or closing of brackets for defining the scope. In Python, we use indentation (spaces or Tabs) for the scope that makes the code look clean and impressive.

Simple Mid-level language

- C++ provides a structured approach wherein you can break the problem into parts and design the solution modularly. It provides you a rich set of library functions that you can use while implementing the solution. If you have worked with C language, then moving to C++ would be a very smooth transitioning. The syntax is almost similar with minute changes.
- C++ has the ability to do both low-level & high-level programming. This is the reason why C++ is known as a mid-level programming language. When we talk about low-level programming, C++ is used to develop system applications such as the kernel, driver, etc.
- C++ is a structured and procedural programming language that makes the execution smooth and fast.

Python & CPP Features

Interpreted high level language

- Python code is not compiled at once, converted to a .exe file, and then executed. Python is an interpreted language which means its code is executed line by line and not all at once like in other programming languages. This line-by-line execution also makes it easy to debug the code.
- Python is a high-level programming language which means that users can easily write and understand or interpret the code. High-level programming language enables the programmer to write codes that are less independent of the specific machine type.
- Python with very strong abstraction from the low-level constructs of the system or machine. While writing code, the developer need not be concerned about the architecture, memory management, or the underlying machine type.

Compiler based language

- C++ is a compiler-based programming language. Without compilation, no C++ program can be executed. The compiler first compiles the C++ program and then it is executed. Without compilation, there is no execution. Being compiler-based, it is comparatively faster than interpreter-based programming languages like Java and Python.
- C++ is a fast language as compilation and execution time is less. Also, it has a wide variety of data types, functions & operators. Being a compiler-based language, C++ is much faster than many programming languages. Compilation and execution time are also faster allowing it to build game engines. C++ execution time and compiling time are faster than any other programming language.

Python & CPP Features

Cross-Platform Language

- Python is a cross-platform language. Many times while downloading some software from some website you might have noticed a list of versions of that software compatible with different operating systems.
- In Python it's not the case, once you write a python code on one machine or operating system then it can be run anywhere. For example, if we have made a python program on Mac then we can run the same code in Linux, Windows, or any other operating system without any changes. This is because the python code first is converted to an intermediate form known as Bytecode and is then executed.

Platform Dependent

- Platform dependent language means the language in which programs can be executed only on that operating system where it is developed & compiled. It cannot run or execute it on any other operating system. C++ is a platform-dependent language. C++ programs can be executed in many machines.
- C++ is machine-independent, which means programs can be executed on many machines with little or no changes. However C++ is not platform-independent, so programs can only be executed on the platform on which they were developed & compiled. The compiler produces an .exe file that is OS-dependent, and cannot be run on different operating systems. The executable file of the C++ program cannot run on different Operating systems/platforms.

Python & CPP Features

Dynamic Typed Language

- In Python, we need not declare the data type of a variable explicitly. The data type of the variable is decided at the run time. Apart from this one variable can be used to store different types of data at different instances in the program. This feature of python saves a lot of time and helps us to avoid pitfalls that might have occurred if it required the datatype to be mentioned explicitly.

Memory Management

- C++ supports dynamic memory allocation. You can free the allocated memory at any time. Not only this C++ also provides dynamic memory management techniques.
- Memory Management is nothing but assigning memory space to the programs for the betterment of performance. C++ supports dynamic memory allocation, which supports freeing and allocating memory anytime. C++ uses new and delete unary operators to allocate and free memory respectively. We can even call the free() function to free memory anytime.
- While working with classes and objects, constructors and destructors can be used for the same purpose. C++ supports pointer and provides solutions to lots of problems that demand access to memory location. There should be no doubt that C++ is popular.

Python & CPP Features

Object-Oriented Language

- OOP is a programming paradigm that is based on the concepts of classes and objects. Classes serve as a blueprint for objects, which contains the data and methods that act on that data. The object-oriented programming concept is focused on making reusable code with a good level of abstraction.
- Python supports object-oriented programming constructs like classes, data encapsulation, inheritance, polymorphism, etc. In Python, we can easily create and use classes, objects and can implement OOP constructs. Due to this approach, one can build efficient and powerful applications in Python.

Object-Oriented Language

- C++ is an object-oriented language, unlike C which is a procedural language. This is one of the most important features of C++. It employs the use of objects while programming. These objects help you implement real-time problems based on data abstraction, data encapsulation, data hiding, and polymorphism.
- In C++ programming, the code is modular with the help of functions, classes & objects, and the modules are loosely coupled. Modular code is easy to understand & modify. This makes C++ a structured programming language. We know that a subset of the procedural programming language is a structured programming language.
- It allows users to break the whole program into smaller units or functions. This programming paradigm aims at improving the readability, reusability, and clarity of the code.

Python & CPP Features

Extensive feature

- Python has the capabilities to be extended and be a more versatile programming language. Python proves to be a versatile language as it covers a large area in software development applications due to its adaptability to various functionalities.
- We can compile the code in languages like C/C++, and then can use that in our python code which can be compiled and run anywhere. It allows the execution of the code written in other programming languages. This provides Python new capabilities and functionality by integrating other programming language's code.

Integration and Extensibility

- Object-oriented support enables C++ programs to be maintainable and extensible. This means that large scale applications can be created. C++ has the potential to integrate and apply newer features easily. It has been used effectively for many diverse applications, including mobile app and game development, software development, and web browser development.
- Many tools in C++, such as style and coding convention checkers, code optimizers, program visualizers and incremental compilers, rely on a continually updated database that contains semantic information extracted from source programs. Additionally, new programming techniques create a need for an extension. C++ has the potential to adopt and integrate newer features and knowledge acquisition easily.

Python & CPP Features

GUI Programming Support

- Graphical User Interface is sometimes shortened to GUI. The user chooses an option – usually by pointing a mouse at an icon representing that option, then clicking to select.
- Features of a GUI include Much easier to use for beginners, Enable an easy exchange of information between software using cut and paste, or 'drag and drop', Use a lot of memory and processing power (can be slower than command line interfaces operated by expert users), Can be irritating to experienced users when simple tasks require a number of operations.
- Python has many GUI libraries like Tkinter, PyQT5, PSide2, etc.

GUI Programming Support

- Graphical User Interface is sometimes shortened to GUI. The user chooses an option – usually by pointing a mouse at an icon representing that option, then clicking to select.
- Features of a GUI include Much easier to use for beginners, Enable an easy exchange of information between software using cut and paste, or 'drag and drop', Use a lot of memory and processing power (can be slower than command line interfaces operated by expert users), Can be irritating to experienced users when simple tasks require a number of operations.
- C++ has many GUI libraries like Qt, Sciter, GTK+, gtkmm., etc.

Python & CPP Features

Large Standard Library

- Python consists of a bulk of libraries that are cross-platform and provides a rich set of modules and functions. These libraries are compatible with various operating systems like UNIX, Mac, windows, etc. Due to the large number of libraries we need not write code for every single thing instead import and use the functionality required.
- For example, if you need to access some websites and want to scrape data from them, then you don't need to write the functions for request, response, and other things, from scratch. There are various libraries available for this purpose which you can use. We can also install other packages that aren't a part of the standard library if we need more functionality.

Rich Library

- Developers have access to lots of in-built functions provided by C++ language. This saves time & makes development fast. Let's look at some of the C++ header files & functionalities provided by it. C++ along with other added benefits provides users with a vast range of in-built libraries.
- These libraries help in making the software development process faster and better. Just by the use of proper header files, such libraries can be accessed by everyone.
- It supports a wide range of data types, functions, and operators. As a result, operating systems, browsers, games, and so forth, can be developed. C++ is an advanced language that supports a wide range of programming techniques, like procedural, object-oriented, functional. So, C++ is very powerful and flexible.

Python & CPP
Industries

Python & CPP Industries

Software Development

- Python is widely used in software development, across a wide variety of real-world applications. The line between software development and web development is a bit blurry these days, since almost all software is built to work on the web even when there's also a desktop app. Dropbox is a good example of a modern software development company that does both, and Python was used to build Dropbox's desktop app. Similarly, Spotify has both web and desktop apps, and Python was used to build a number of the background services that make them work.
- Python is great for automating repetitive tasks, and there are almost endless real world use cases for Python automation. It is a popular tool in DevOps because it makes automating systems and processes efficient.

Software Development

- Banks and financial institutions use C++ to build their software infrastructure, which powers applications for banking, trading, and financial modeling. The speed and reliable performance of C++ is ideal for processing millions of daily transactions, facilitating a high volume and frequency of trades, and creating data simulations for large portfolios.
- C++ can be found in a variety of medical applications, from MRI machines to lab testing equipment to systems that handle patient information. C++ is also used to model data and run data simulations for bioinformatics research, because many advanced algorithms written for the medical and biological sciences are implemented in C++.
- C++ is used in building telephone, Internet, and other telecommunications infrastructure.

Python & CPP Industries

Web application development

- Unarguably, one of the top practical uses for Python is web application development. Python is now easily the go-to programming language for web applications.
- Web development has several uses of Python in the real world. It provides security, convenience, and scalability to applications.
- Python has a lot of web development frameworks like Django and Flask, which enable rapid app development. Django's dynamic development capabilities have made Python a useful tool for web applications.
- The framework is packed with standard libraries, reducing the development time and providing more time-to-market for the web application.

Web application development

- C++ plays a role in web browsers, such as Google Chrome, Mozilla Firefox, Safari, and Opera. It is used to develop back-end services that retrieve information from databases and render code into interactive web pages. C++ helps web browsers carry out these tasks with speed and minimal delays, which means we don't have to wait long for content to appear on our screens.
- Modern web browsers need to display complex and dynamic web pages within milliseconds to offer a positive user experience. This requires a high-performance rendering engine, which is most often written in C++. This is the case for Mozilla Firefox,Google Chrome, and most other modern web browsers.

Python & CPP Industries

Machine Learning Tools

- Data science is now reaching the top. It is becoming one of the most important areas with applications of Python programming. Python libraries like Pandas, NumPy, SciPy, and others help you to work with data and extract valuable information and insights.
- Data scientists have to know the uses of Python for extracting and processing data. It allows them to visualize the data through graphs. Matplotlib and Seaborn, both are used for data visualization. It is the first thing that data scientists have to learn. It is preliminary to working with data-based companies.
- Machine Learning algorithms are one of the important real life uses of Python. Developers can write algorithms easily using Python. It has an extensive collection of libraries for ML apps include SciPy, Pandas, Keras, TensorFlow.

Machine Learning Tools

- Machine learning tools need to process massive amounts of data quickly and efficiently, making a fast programming like C++ essential. For example, the core calculations for TensorFlow, a popular ML framework, are written with C++ code.
- Machine learning tools, such as TensorFlow, rely on C++ as a back-end programming language. Even though data scientists can use TensorFlow with Python, for example, the core machine learning calculations are carried out with C++ code. In fact, C++ has a large collection of libraries that power these highly-sophisticated calculations that train machine learning models.

Python & CPP Industries

Game development

- Gaming app development is now a prominent industry, and it has many applications of Python programming. There are libraries which are widely used for interactive game development.
- Some of the real world Python projects in the gaming industry include Battlefield 2, Frets on Fire, World of Tanks, etc. These games use Python libraries like PySoy, PyGame.
- Python allows game developers to build tree-based algorithms which are useful in designing different levels in a game. Games require handling multiple requests at once, and Python is extremely fantastic at that.
- Python game app development is one of the top 10 uses of Python in the real world. It offers developers the opportunity to install a 3D game engine that helps in building powerful games and interfaces.

Game development

- Modern game development requires high performance to render complex 3D worlds and handle multiplayer networking. C++ has features for low-level resource manipulation that allows game developers to make the most of the hardware a game is running on.
- C++ is also a popular option for building augmented and virtual reality (AR/VR) applications, because massive amounts of sensor data must be processed in real time. In addition, many AR/VR games are built atop the Unreal Engine, which is written in C++.
- C++ is one of the most widely-used programming languages in game development. It has been used to create games, such as World of Warcraft, Counter-Strike, and StarCraft, game engines like Unreal Engine, gaming consoles, Xbox, PlayStation, Nintendo Switch.

Python & CPP Industries

Embedded Systems & Internet of Things

- Another one of the real life uses of Python is in the IoT. Python enables developers to turn any object into an electronic gadget with the help of Raspberry Pi. Python is used to create embedded software, allowing high-performance application of Python on smaller objects which can work with the programming language.
- With the help of Raspberry Pi, developers can do high-level computations using Python applications. By embedding it, developers can turn normal objects into smart electronics.
- In large scale industries, IoT is widely used to track inventory, move machines, and track order processing along with the status of shipment. Python's ability to control hardware goes beyond robotics, it is used in all sorts of real world hardware-control applications.

Embedded Systems & Internet of Things

- Embedded systems are devices like smartwatches, medical devices, and other equipment that combines hardware and software into a dedicated solution. Because the software is designed specifically for a particular device, using C++ lower-level language allows developers to use low-level function calls and directly manage resource usage to get the best performance out of the hardware.
- Similar to embedded systems, the software for internet of things (IoT) devices needs to run close to the hardware. That's because IoT integrates numerous, internet-connected devices into a centrally controlled system.
- C++ is ideal for IoT because it requires high-performance, fast networking capabilities, and ability to quickly process large data amounts.

Python & CPP Industries

Desktop GUI

- Python programming language can work with multiple operating systems and has a powerful architecture for building applications.
- It has rich text processing tools and a clear syntax, allowing developers to code Desktop GUI applications without any hassle.
- PyQT, Kivy, PyGUI are a few toolkits and frameworks offered to get you started with the practical uses of Python for GUI development.
- Developers can create highly functional GUIs with Python and reduce the turnaround time for development.

Desktop GUI

- The C++ ecosystem has a variety of frameworks for building applications with graphical user interfaces (GUIs).
- These frameworks or GUI toolkits are often provided by each operating system, so developers can design applications with native user interfaces.
- There are also many cross-platform GUI frameworks for building applications with user interfaces across multiple platforms.

Lecture Agenda

1- Why this Training is Important

2- Training Content and Educational Projects

3- How to Practice on Online Competitions

4- Technical Tutorials and Reference Textbooks

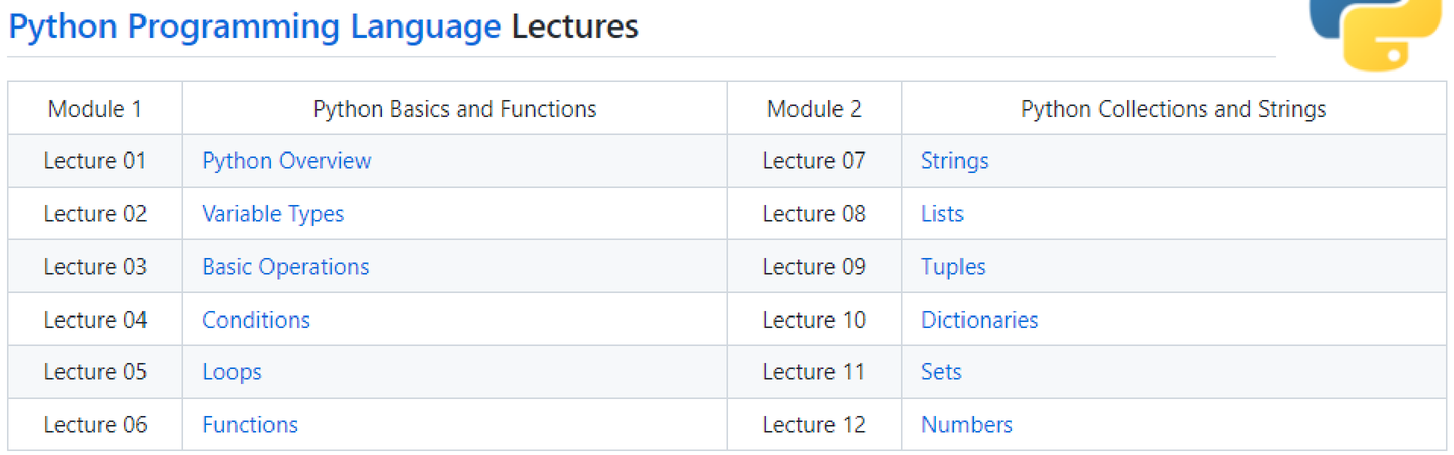
5- Online Learning Platforms Courses

6- Environment Setup and Installations

Training Content

Training Content - Python Programming Language

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/tree/master/Lecture-Notes

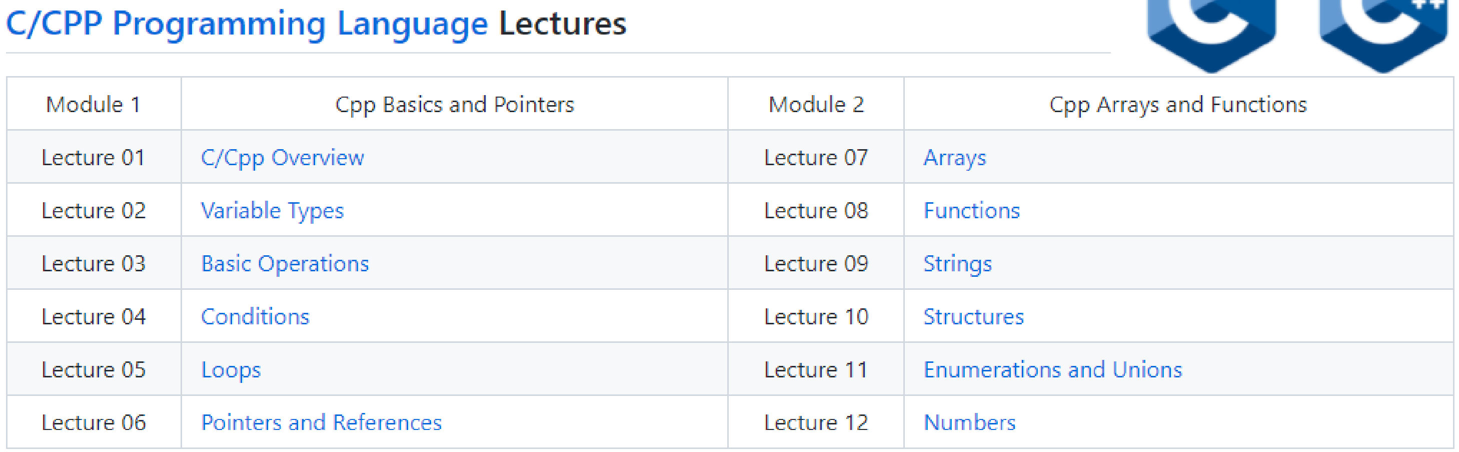


Python Programming Language Lectures

Module 1	Python Basics and Functions	Module 2	Python Collections and Strings
Lecture 01	Python Overview	Lecture 07	Strings
Lecture 02	Variable Types	Lecture 08	Lists
Lecture 03	Basic Operations	Lecture 09	Tuples
Lecture 04	Conditions	Lecture 10	Dictionaries
Lecture 05	Loops	Lecture 11	Sets
Lecture 06	Functions	Lecture 12	Numbers

Training Content - CPP Programming Language

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/tree/master/Lecture-Notes

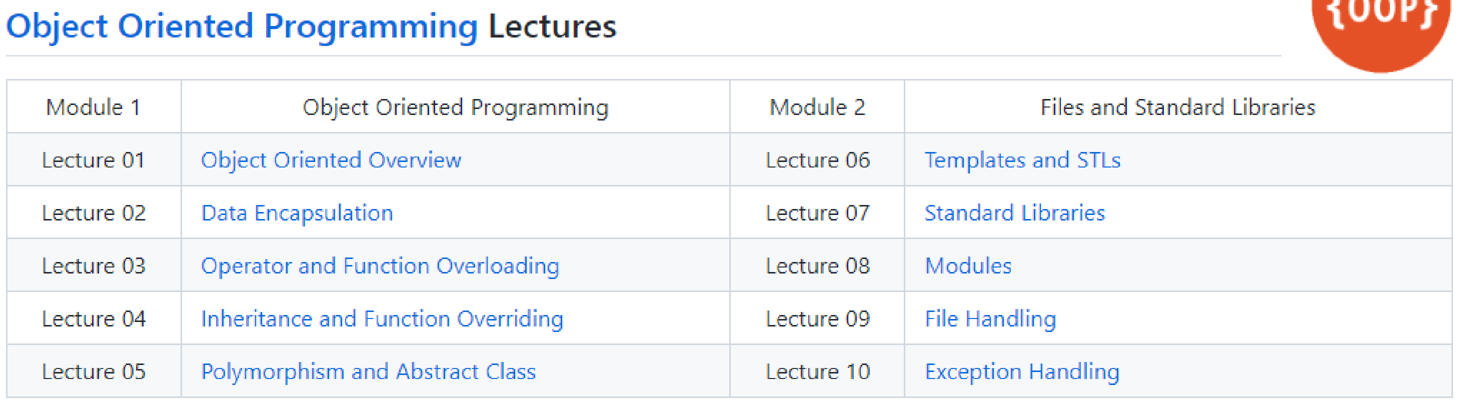


C/CPP Programming Language Lectures

Module 1	Cpp Basics and Pointers	Module 2	Cpp Arrays and Functions
Lecture 01	C/Cpp Overview	Lecture 07	Arrays
Lecture 02	Variable Types	Lecture 08	Functions
Lecture 03	Basic Operations	Lecture 09	Strings
Lecture 04	Conditions	Lecture 10	Structures
Lecture 05	Loops	Lecture 11	Enumerations and Unions
Lecture 06	Pointers and References	Lecture 12	Numbers

Training Content - Object-Oriented Programming

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/tree/master/Lecture-Notes

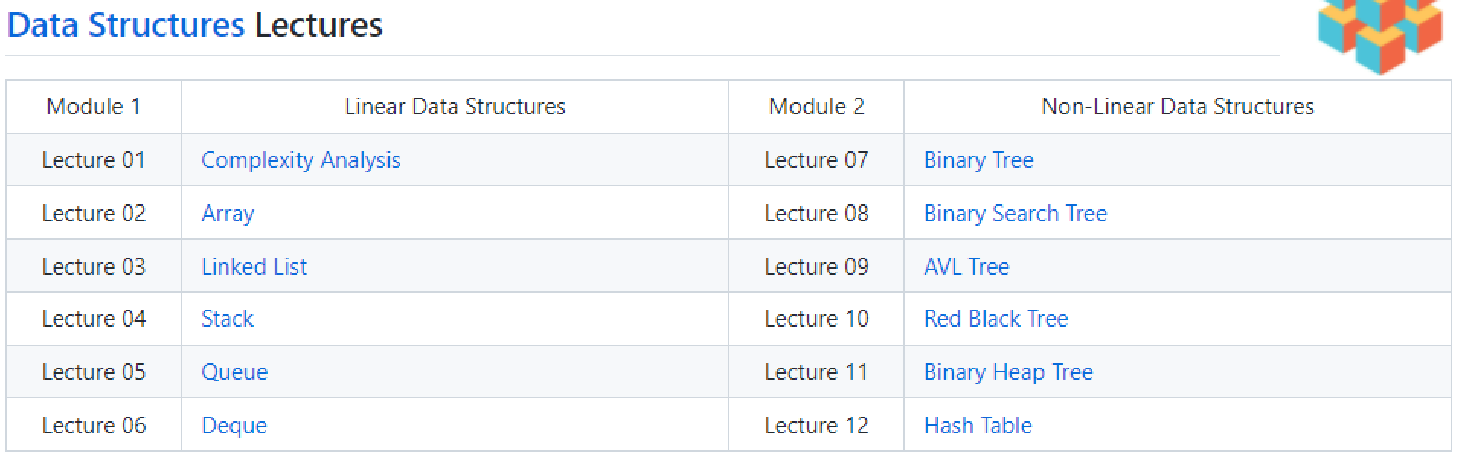


Object Oriented Programming Lectures

Module 1	Object Oriented Programming	Module 2	Files and Standard Libraries
Lecture 01	Object Oriented Overview	Lecture 06	Templates and STLs
Lecture 02	Data Encapsulation	Lecture 07	Standard Libraries
Lecture 03	Operator and Function Overloading	Lecture 08	Modules
Lecture 04	Inheritance and Function Overriding	Lecture 09	File Handling
Lecture 05	Polymorphism and Abstract Class	Lecture 10	Exception Handling

Training Content - Data Structures

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/tree/master/Lecture-Notes

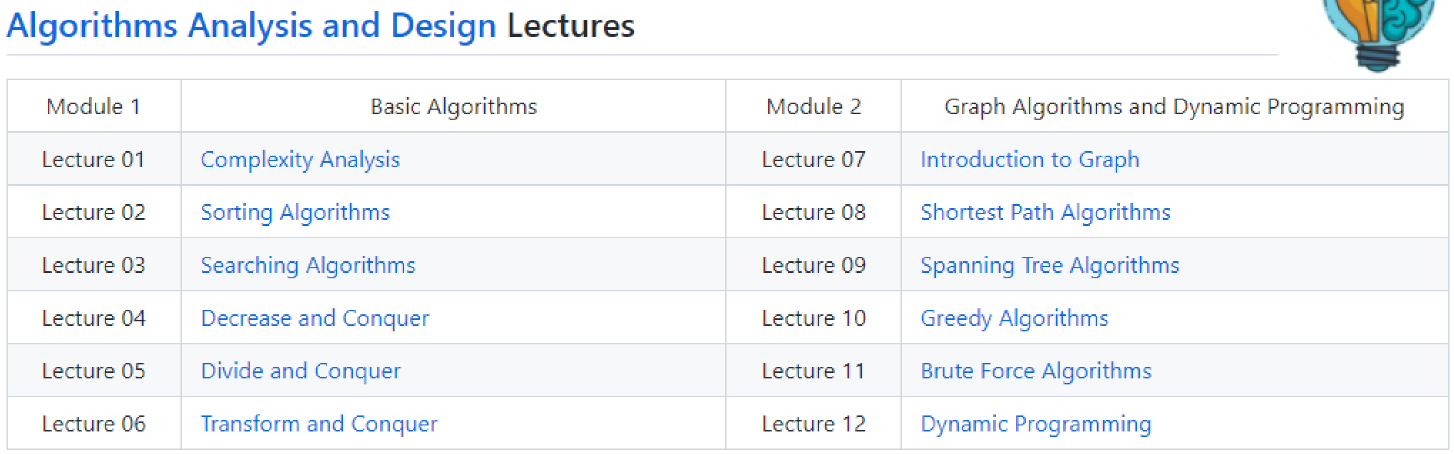


Data Structures Lectures

Module 1	Linear Data Structures	Module 2	Non-Linear Data Structures
Lecture 01	Complexity Analysis	Lecture 07	Binary Tree
Lecture 02	Array	Lecture 08	Binary Search Tree
Lecture 03	Linked List	Lecture 09	AVL Tree
Lecture 04	Stack	Lecture 10	Red Black Tree
Lecture 05	Queue	Lecture 11	Binary Heap Tree
Lecture 06	Deque	Lecture 12	Hash Table

Training Content - Algorithms Analysis and Design

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/tree/master/Lecture-Notes



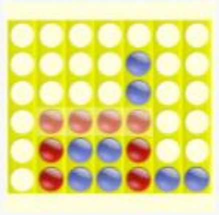
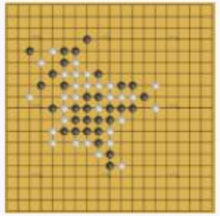
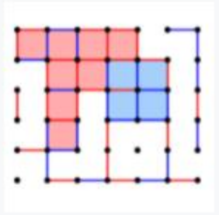
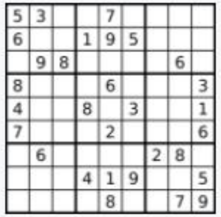
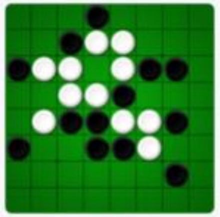
Algorithms Analysis and Design Lectures

Module 1	Basic Algorithms	Module 2	Graph Algorithms and Dynamic Programming
Lecture 01	Complexity Analysis	Lecture 07	Introduction to Graph
Lecture 02	Sorting Algorithms	Lecture 08	Shortest Path Algorithms
Lecture 03	Searching Algorithms	Lecture 09	Spanning Tree Algorithms
Lecture 04	Decrease and Conquer	Lecture 10	Greedy Algorithms
Lecture 05	Divide and Conquer	Lecture 11	Brute Force Algorithms
Lecture 06	Transform and Conquer	Lecture 12	Dynamic Programming

Educational Projects

Educational Projects - Programming Languages

Repository Link: github.com/cs-MohamedAyman/Educational-Projects

Tic Tac Toe Game	Connect Four Game	Gomoku Game	8 Puzzle Game	SOS Game	Dots and Boxes Game
					
Snakes and Ladders Game	Sudoku Game	Match Cards Game	Reversi Game	Draughts Game	Minesweeper Game
					

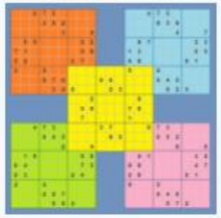
Educational Projects - Object-Oriented

Repository Link: github.com/cs-MohamedAyman/Educational-Projects

Matrix Calculator	Company System	Building System	Library System	Bank System	Champions League System
Hospital System	Radio System	University System	Cinema System	E-Commerce System	Cooking System

Educational Projects - Data Structures & Algorithms

Repository Link: github.com/cs-MohamedAyman/Educational-Projects

Mobile Contacts System	Restaurant Orders System	Issue Tracking System	Transportation Ticketing System	Equation Solver	2048 Game
Ultimate Tic Tac Toe Game	Battleship Game	Multi Sudoku Game	Zuma Game	Snake Game	Ball Sort Game
					

Educational Projects - Data Structures & Algorithms

Repository Link: github.com/cs-MohamedAyman/Educational-Projects

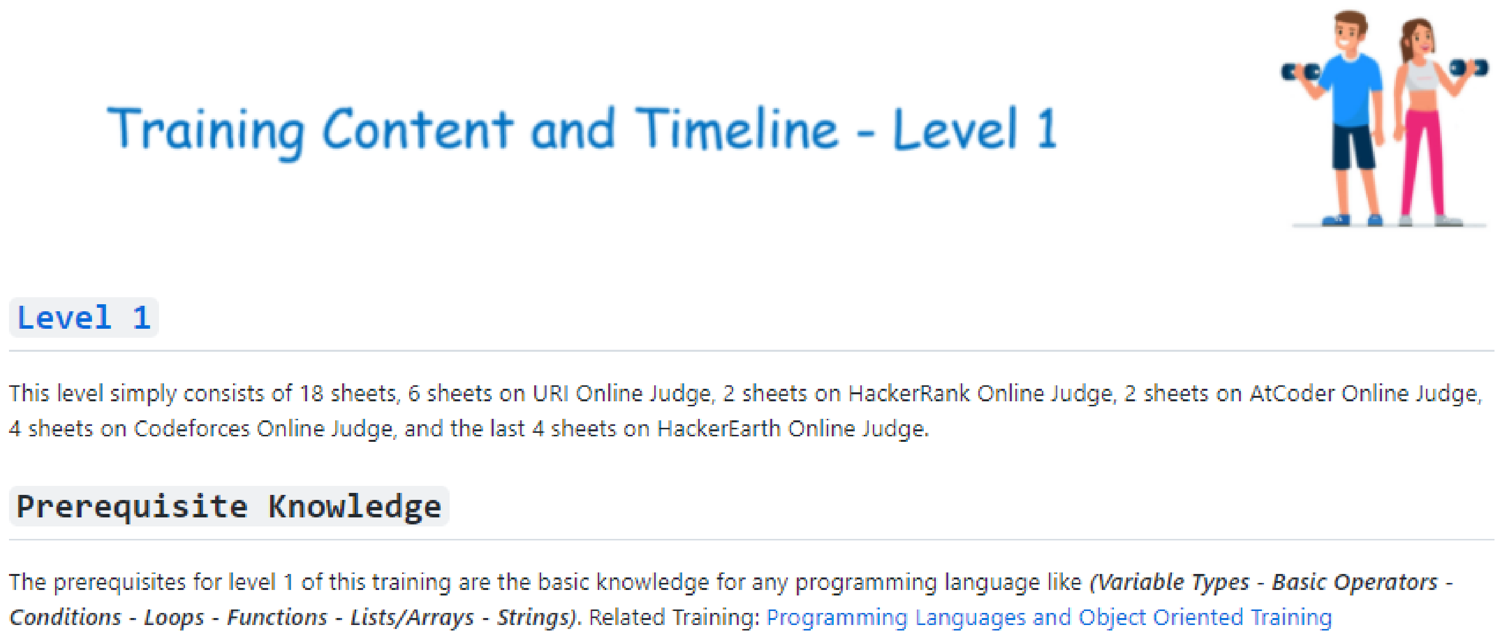
Merge Block Game	Word Search Game	Hangman Game	Bingo Game	Maze Game	Pacman Game
8 Queens Game	Knights Tour Game	Chess Endgame Game	Chess Game	Fruit Master Game	Bubble Poke Game

Lecture Agenda

Problem Solving Training Level 1

Problem Solving Training - Level 1

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training



Training Content and Timeline - Level 1

Level 1

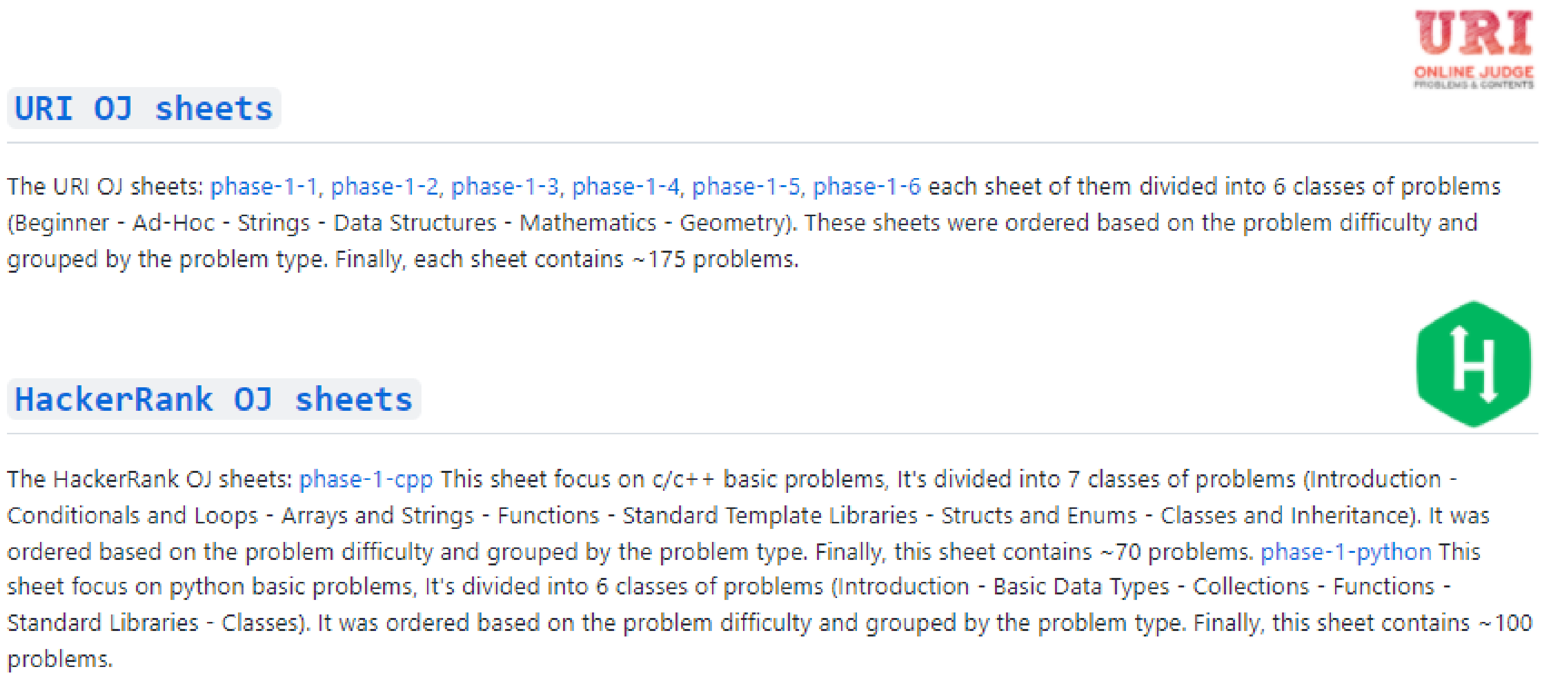
This level simply consists of 18 sheets, 6 sheets on URI Online Judge, 2 sheets on HackerRank Online Judge, 2 sheets on AtCoder Online Judge, 4 sheets on Codeforces Online Judge, and the last 4 sheets on HackerEarth Online Judge.

Prerequisite Knowledge

The prerequisites for level 1 of this training are the basic knowledge for any programming language like ***(Variable Types - Basic Operators - Conditions - Loops - Functions - Lists/Arrays - Strings)***. Related Training: Programming Languages and Object Oriented Training

Problem Solving Training - Level 1 sheets

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training



URI OJ sheets

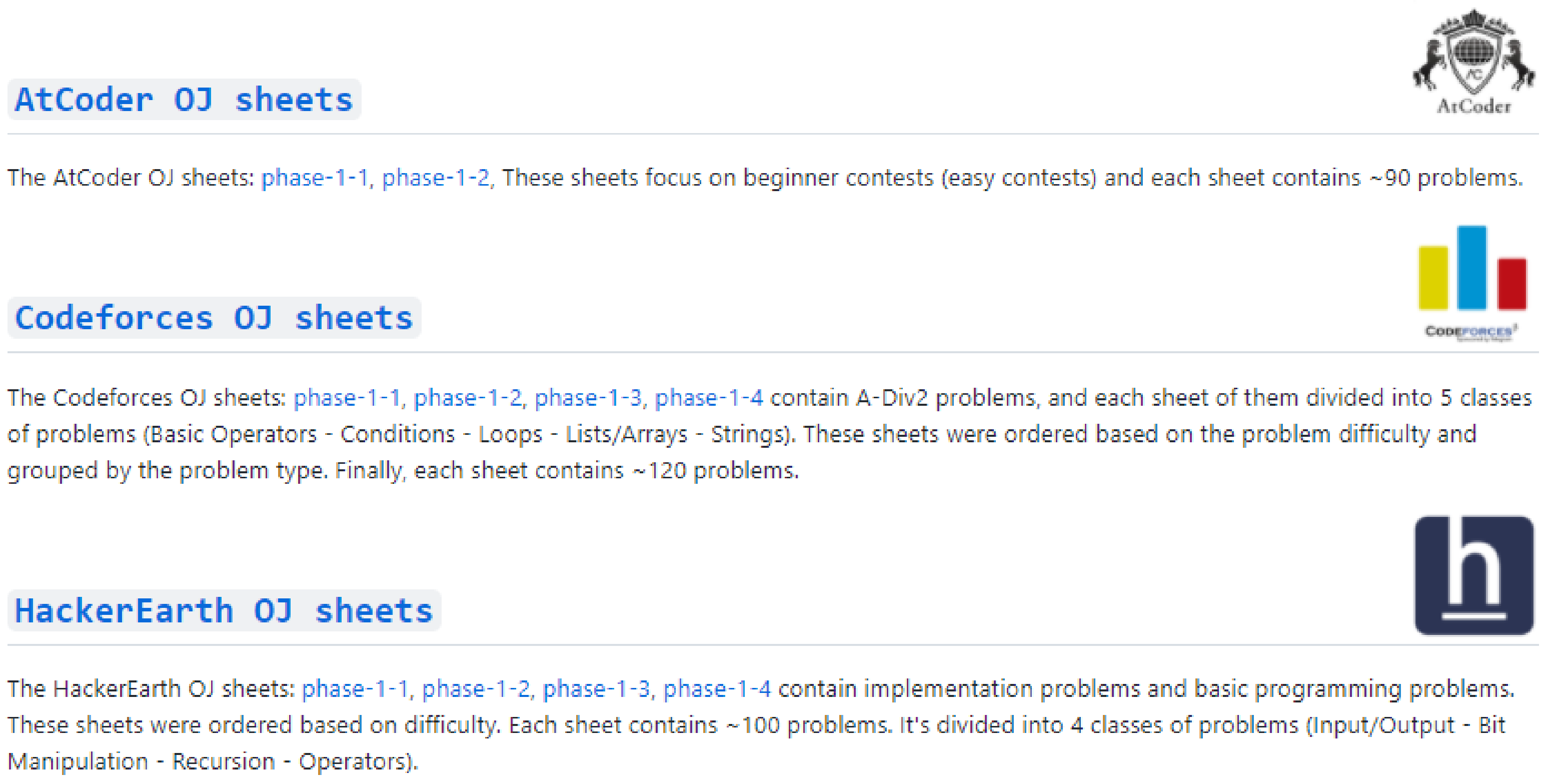
The URI OJ sheets: phase-1-1, phase-1-2, phase-1-3, phase-1-4, phase-1-5, phase-1-6 each sheet of them divided into 6 classes of problems (Beginner - Ad-Hoc - Strings - Data Structures - Mathematics - Geometry). These sheets were ordered based on the problem difficulty and grouped by the problem type. Finally, each sheet contains ~175 problems.

HackerRank OJ sheets

The HackerRank OJ sheets: phase-1-cpp This sheet focus on c/c++ basic problems, It's divided into 7 classes of problems (Introduction - Conditionals and Loops - Arrays and Strings - Functions - Standard Template Libraries - Structs and Enums - Classes and Inheritance). It was ordered based on the problem difficulty and grouped by the problem type. Finally, this sheet contains ~70 problems. phase-1-python This sheet focus on python basic problems, It's divided into 6 classes of problems (Introduction - Basic Data Types - Collections - Functions - Standard Libraries - Classes). It was ordered based on the problem difficulty and grouped by the problem type. Finally, this sheet contains ~100 problems.

Problem Solving Training - Level 1 sheets

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training



AtCoder OJ sheets

The AtCoder OJ sheets: phase-1-1, phase-1-2, These sheets focus on beginner contests (easy contests) and each sheet contains ~90 problems.

Codeforces OJ sheets

The Codeforces OJ sheets: phase-1-1, phase-1-2, phase-1-3, phase-1-4 contain A-Div2 problems, and each sheet of them divided into 5 classes of problems (Basic Operators - Conditions - Loops - Lists/Arrays - Strings). These sheets were ordered based on the problem difficulty and grouped by the problem type. Finally, each sheet contains ~120 problems.

HackerEarth OJ sheets

The HackerEarth OJ sheets: phase-1-1, phase-1-2, phase-1-3, phase-1-4 contain implementation problems and basic programming problems. These sheets were ordered based on difficulty. Each sheet contains ~100 problems. It's divided into 4 classes of problems (Input/Output - Bit Manipulation - Recursion - Operators).

Problem Solving Training - Level 1 Timeline

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training

Day 01-10	URI ONLINE JUDGE		URI phase-1-1
			HackerRank phase-1-cpp
Day 11-20	CODEFORCES		Codeforces phase-1-1
			HackerEarth phase-1-1
Day 21-30	URI ONLINE JUDGE		URI phase-1-2
			HackerRank phase-1-python
Day 31-40	CODEFORCES		Codeforces phase-1-2
			HackerEarth phase-1-2
Day 41-50	URI ONLINE JUDGE	AtCoder	URI phase-1-3
			AtCoder phase-1-1
Day 51-60	CODEFORCES		Codeforces phase-1-3
			HackerEarth phase-1-3

Problem Solving Training - Level 1 Timeline

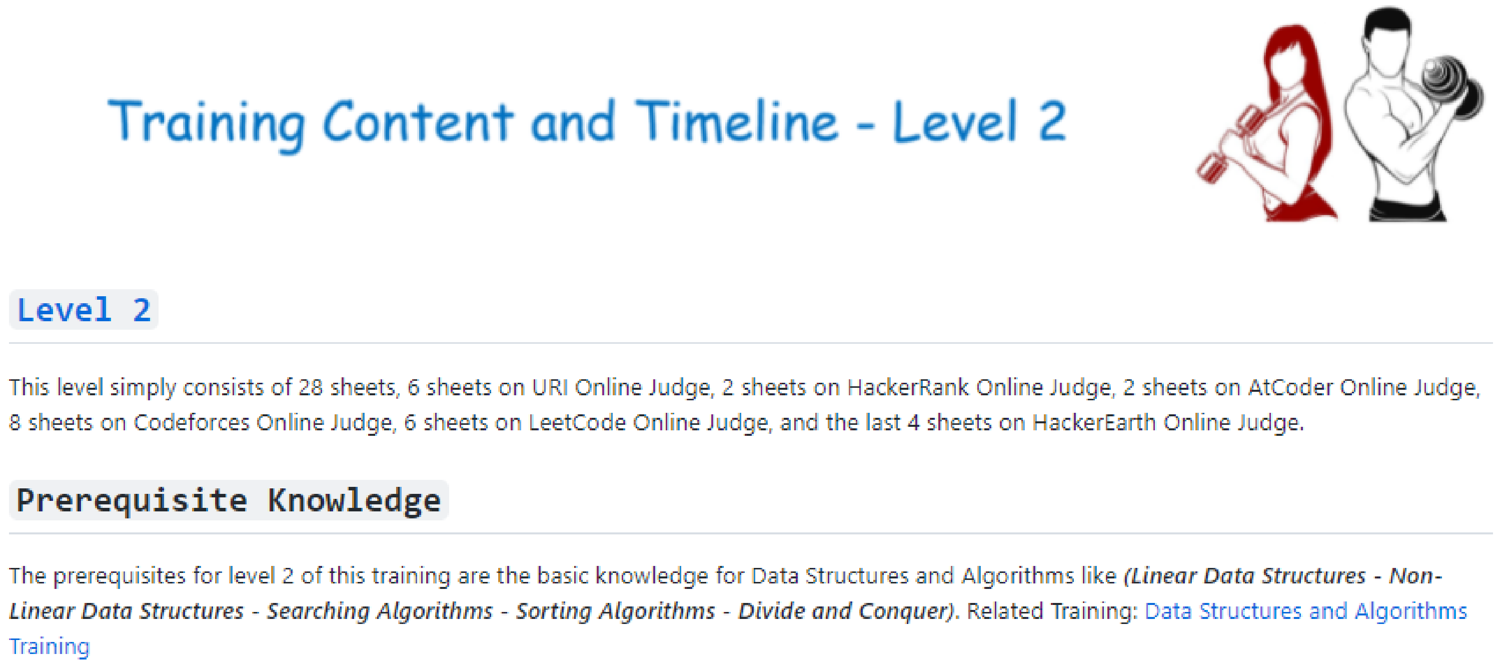
Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training

Day 61-70	URI ONLINE JUDGE	AtCoder	URI phase-1-4
			AtCoder phase-1-2
Day 71-80	CODEFORCES		Codeforces phase-1-4
			HackerEarth phase-1-4
Day 81-90	URI ONLINE JUDGE	URI ONLINE JUDGE	URI phase-1-5
			URI phase-1-6

Problem Solving Training Level 2

Problem Solving Training - Level 2

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training



Training Content and Timeline - Level 2

Level 2

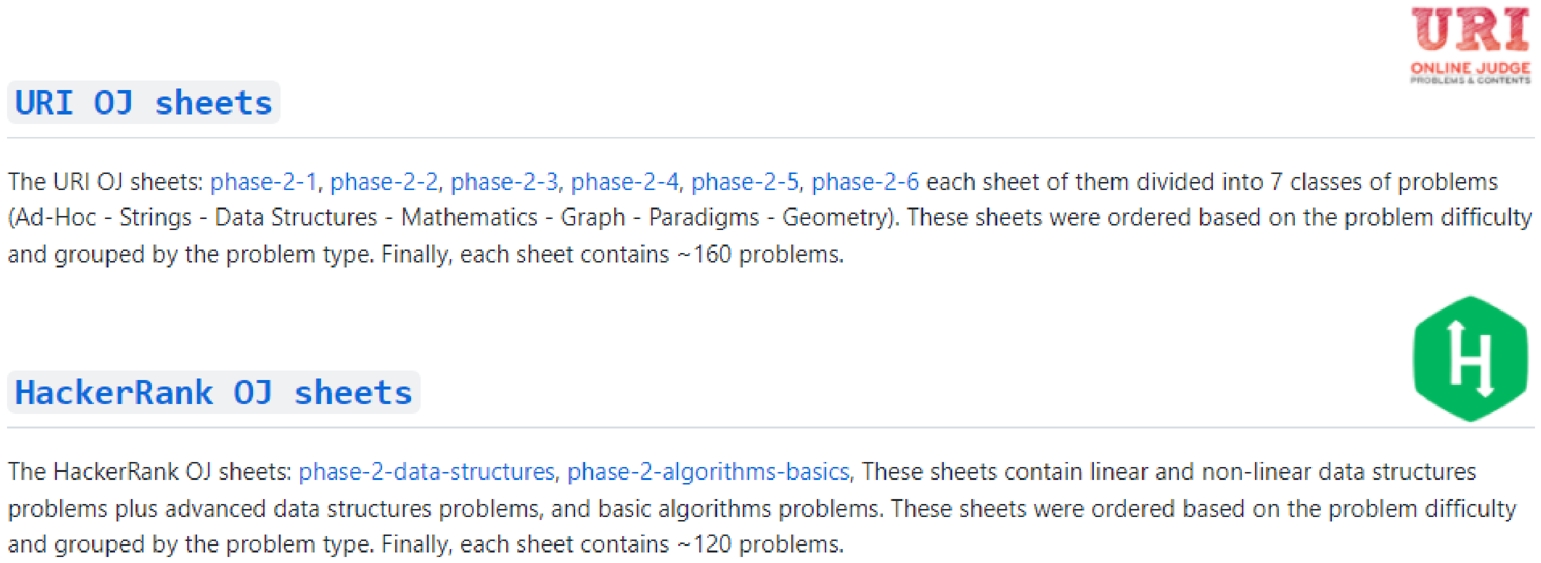
This level simply consists of 28 sheets, 6 sheets on URI Online Judge, 2 sheets on HackerRank Online Judge, 2 sheets on AtCoder Online Judge, 8 sheets on Codeforces Online Judge, 6 sheets on LeetCode Online Judge, and the last 4 sheets on HackerEarth Online Judge.

Prerequisite Knowledge

The prerequisites for level 2 of this training are the basic knowledge for Data Structures and Algorithms like ***(Linear Data Structures - Non-Linear Data Structures - Searching Algorithms - Sorting Algorithms - Divide and Conquer)***. Related Training: Data Structures and Algorithms Training

Problem Solving Training - Level 2 sheets

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training



URI OJ sheets

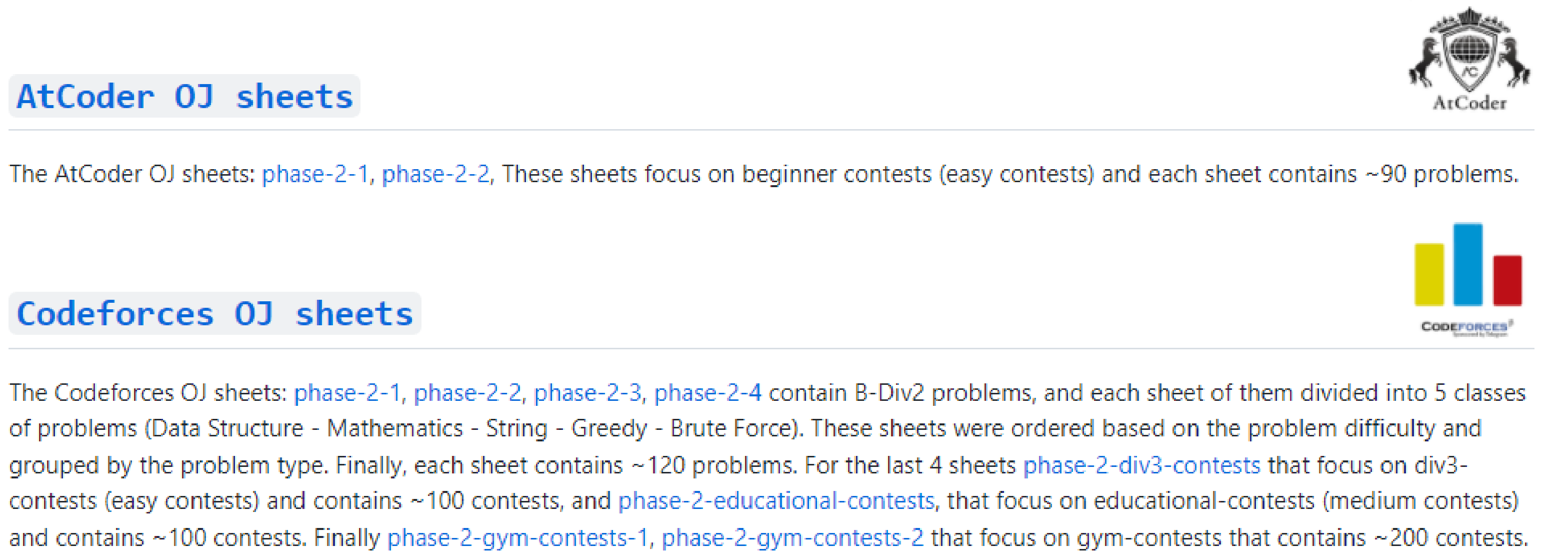
The URI OJ sheets: phase-2-1, phase-2-2, phase-2-3, phase-2-4, phase-2-5, phase-2-6 each sheet of them divided into 7 classes of problems (Ad-Hoc - Strings - Data Structures - Mathematics - Graph - Paradigms - Geometry). These sheets were ordered based on the problem difficulty and grouped by the problem type. Finally, each sheet contains ~160 problems.

HackerRank OJ sheets

The HackerRank OJ sheets: phase-2-data-structures, phase-2-algorithms-basics, These sheets contain linear and non-linear data structures problems plus advanced data structures problems, and basic algorithms problems. These sheets were ordered based on the problem difficulty and grouped by the problem type. Finally, each sheet contains ~120 problems.

Problem Solving Training - Level 2 sheets

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training



AtCoder OJ sheets

The AtCoder OJ sheets: phase-2-1, phase-2-2, These sheets focus on beginner contests (easy contests) and each sheet contains ~90 problems.

Codeforces OJ sheets

The Codeforces OJ sheets: phase-2-1, phase-2-2, phase-2-3, phase-2-4 contain B-Div2 problems, and each sheet of them divided into 5 classes of problems (Data Structure - Mathematics - String - Greedy - Brute Force). These sheets were ordered based on the problem difficulty and grouped by the problem type. Finally, each sheet contains ~120 problems. For the last 4 sheets phase-2-div3-contests that focus on div3-contests (easy contests) and contains ~100 contests, and phase-2-educational-contests, that focus on educational-contests (medium contests) and contains ~100 contests. Finally phase-2-gym-contests-1, phase-2-gym-contests-2 that focus on gym-contests that contains ~200 contests.

Problem Solving Training - Level 2 sheets

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training

LeetCode OJ sheets

The LeetCode OJ sheets: phase-2-1, phase-2-2, phase-2-3, phase-2-4, phase-2-5, phase-2-6 contain linear and non-linear data structures problems, searching and sorting algorithms, and each sheet of them divided into 6 classes of problems (Array - LinkedList - Stack - Queue - Binary Tree - Heap Tree - HashTable) in addition to (Binary Search - Sorting - Divide and Conquer - Greedy - Bit Manipulation). These sheets were ordered based on the problem difficulty and grouped by the problem type. Finally, each sheet contains ~190 problems.

HackerEarth OJ sheets

The HackerEarth OJ sheets: phase-2-linear-data-structures, phase-2-non-linear-data-structures, phase-2-algorithms-searching, phase-2-algorithms-sorting, Each sheet contains linear and non-linear data structures problems, in addition to searching and sorting algorithms. These sheets were ordered based on the problem difficulty and each sheet contains ~100 problems.

Problem Solving Training - Level 2 Timeline

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training

Day 01-10	URI ONLINE JUDGE		URI phase-2-1
			HackerRank phase-2-data-structures
Day 11-20	CODEFORCES		Codeforces phase-2-1
			LeetCode phase-2-1
Day 21-30	URI ONLINE JUDGE		URI phase-2-2
			HackerRank phase-2-algorithms-basics
Day 31-40	CODEFORCES		Codeforces phase-2-2
			LeetCode phase-2-2
Day 41-50	URI ONLINE JUDGE	AtCoder	URI phase-2-3
			AtCoder phase-2-1
Day 51-60	CODEFORCES		Codeforces phase-2-3
			LeetCode phase-2-3

Problem Solving Training - Level 2 Timeline

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training

Day 61-70	URI ONLINE JUDGE	AtCoder	URI phase-2-4
			AtCoder phase-2-2
Day 71-80	CODEFORCES		Codeforces phase-2-4
			LeetCode phase-2-4
Day 81-90	URI ONLINE JUDGE	h	URI phase-2-5
			HackerEarth phase-2-linear-data-structures
Day 91-100		h	LeetCode phase-2-5
			HackerEarth phase-2-non-linear-data-structures
Day 101-110	URI ONLINE JUDGE	h	URI phase-2-6
			HackerEarth phase-2-algorithms-searching
Day 111-120		h	LeetCode phase-2-6
			HackerEarth phase-2-algorithms-sorting

Practice on Online Competitions

Repository Link: github.com/cs-MohamedAyman/Problem-Solving-Training

acm International Collegiate Programming Contest

facebook HACKERCUP

code jam print "hello, world!"

Hash Code

[topcoder]™

kick start

Lecture Agenda

1- Why this Training is Important

2- Training Content and Educational Projects

3- How to Practice on Online Competitions

4- Technical Tutorials and Reference Textbooks

5- Online Learning Platforms Courses

6- Environment Setup and Installations

Technical Tutorials

Technical Tutorials - Python and Object-Oriented

Repository Link: github.com/cs-MohamedAyman/Technical-Tutorials

programiz.com/python-programming

docs.python.org/3

geeksforgeeks.org/python-programming-language

tutorialspoint.com/python3

Technical Tutorials - CPP and Object-Oriented

Repository Link: github.com/cs-MohamedAyman/Technical-Tutorials

programiz.com/cpp-programming

cplusplus.com/doc/tutorial

geeksforgeeks.org/c-plus-plus

tutorialspoint.com/cplusplus

Technical Tutorials - Data Structures

Repository Link: github.com/cs-MohamedAyman/Technical-Tutorials

programiz.com/dsa

hackerearth.com/practice/data-structures

geeksforgeeks.org/data-structures

tutorialspoint.com/data_structures_algorithms

Technical Tutorials – Algorithms Analysis and Design

Repository Link: github.com/cs-MohamedAyman/Technical-Tutorials

programiz.com/dsa

hackerearth.com/practice/algorithms

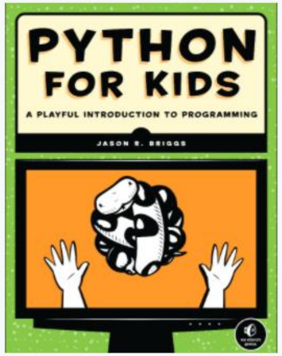
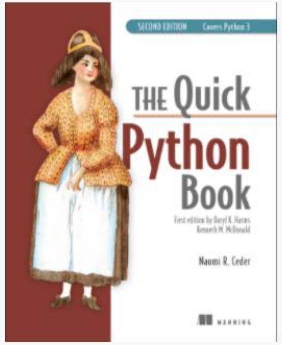
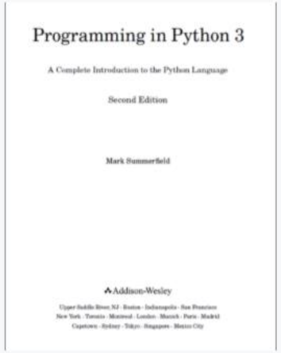
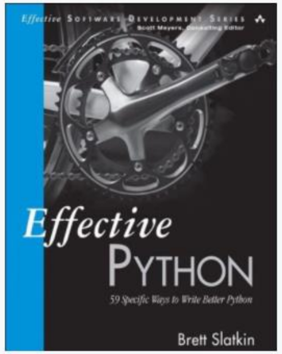
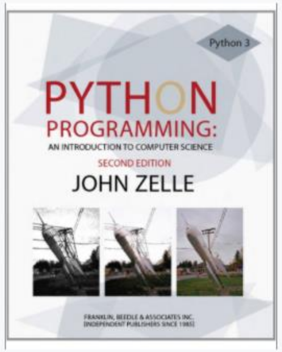
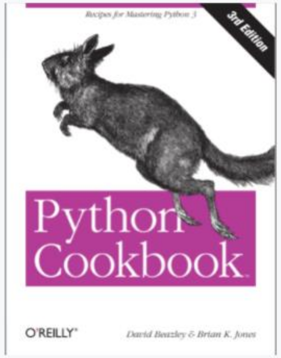
geeksforgeeks.org/fundamentals-of-algorithms

tutorialspoint.com/data_structures_algorithms

Reference Textbooks

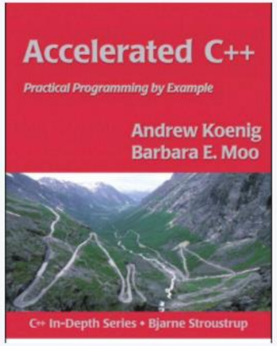
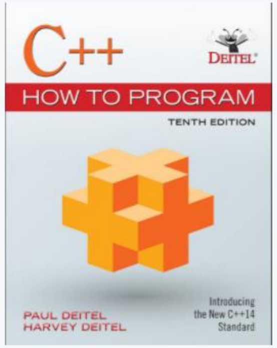
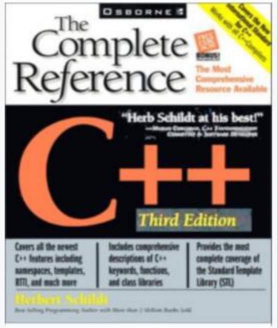
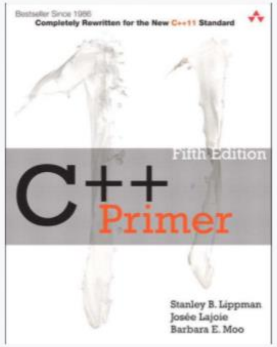
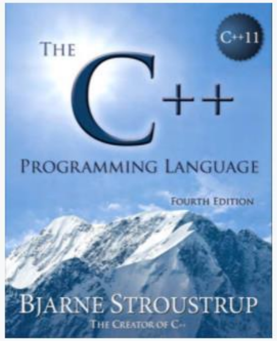
Reference Textbooks - Python Programming Language

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/blob/master/Reference-Textbooks

Python for Kids, Jason R. Briggs	The Quick Python Book Naomi R. Ceder	Programming in Python3, Mark Summerfield
		
Effective Python, Brett Slatkin	Python Programming, John Zelle	Python Cookbook, David Beazley
		

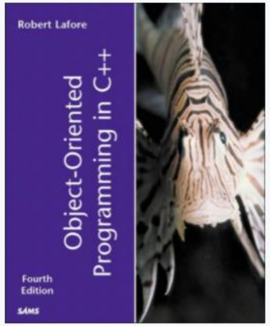
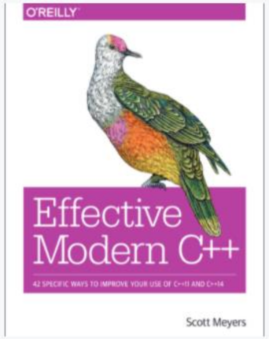
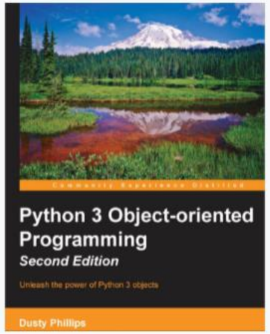
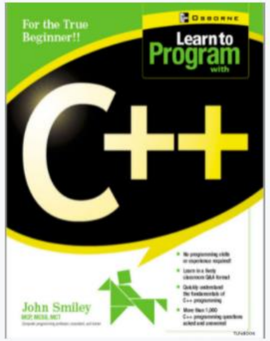
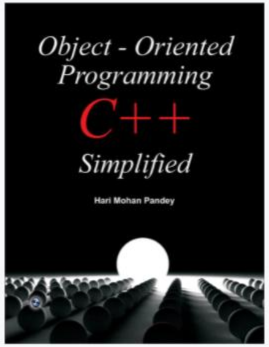
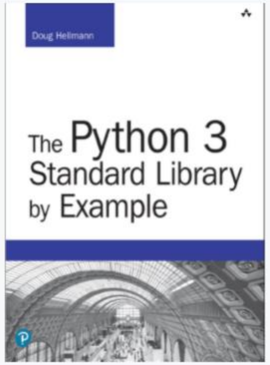
Reference Textbooks - CPP Programming Language

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/blob/master/Reference-Textbooks

A Complete Guide to Programming in Cpp, Ulla Kirch Prinz	Accelerated Cpp, Andrew Koenig and Barbara E. Moo	Cpp How to Program, Paul Deitel
A Complete Guide to Programming in C++ Ulla Kirch-Prinz Peter Prinz	 Accelerated C++ Practical Programming by Example Andrew Koenig Barbara E. Moo C++ In-Depth Series • Bjarne Stroustrup	 C++ DEITEL HOW TO PROGRAM TENTH EDITION PAUL DEITEL HARVEY DEITEL Introducing the New C++14 Standard
Cpp The Complete Reference, Herbert Schildt	Cpp Primer, Stanley B. Lippman	The Cpp Programming Language, Bjarne Stroustrup
 OSBORNE The Complete Reference C++ Third Edition "Herb Schildt at his best!" Herbert Schildt	 Completely Rewritten for the New C++11 Standard C++ Primer Fifth Edition Stanley B. Lippman Josée Lajoie Barbara E. Moo	 C++11 THE C++ PROGRAMMING LANGUAGE FOURTH EDITION BJARNE STROUSTRUP THE CREATOR OF C++

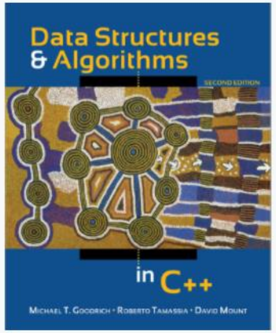
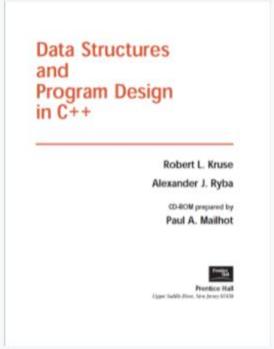
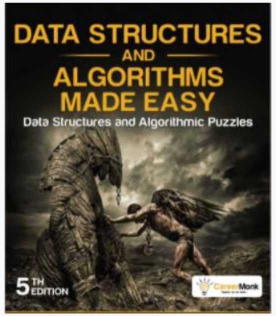
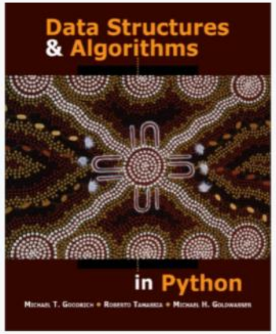
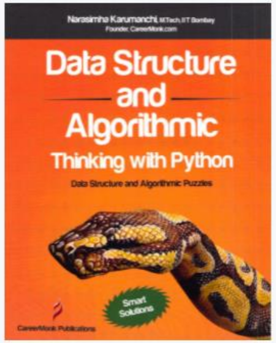
Reference Textbooks - Object-Oriented

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/blob/master/Reference-Textbooks

Object-Oriented Programming in C++, Robert Lafore	Effective Modern Cpp, Scott Meyers	Python3 Object Oriented Programming, Dusty Phillips
 Robert Lafore Object-Oriented Programming in C++ Fourth Edition SAMS	 O'REILLY Effective Modern C++ 42 SPECIFIC WAYS TO IMPROVE YOUR USE OF C++11 AND C++14 Scott Meyers	 Python 3 Object-oriented Programming Second Edition Unleash the power of Python 3 objects Dusty Phillips
Learn to Program with Cpp, John Smiley	Object Oriented Programming C++ Simplified, Hari Mohan-Pandey	The Python3 Standard Library by Example, Doug Hellmann
 For the True Beginner!! OSBORNE Learn to Program with C++ John Smiley	 Object - Oriented Programming C++ Simplified Hari Mohan Pandey	

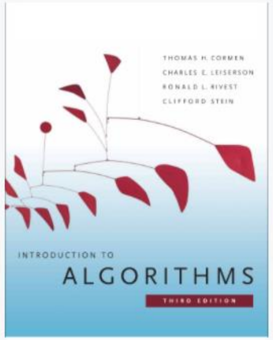
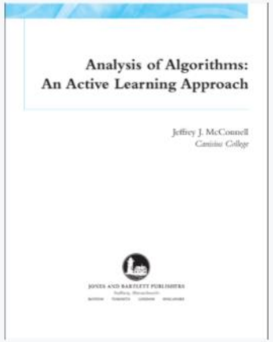
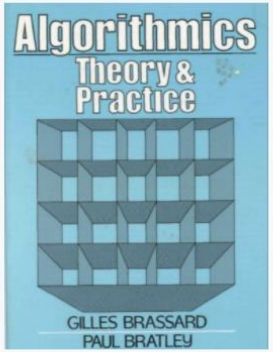
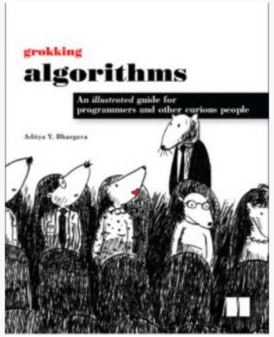
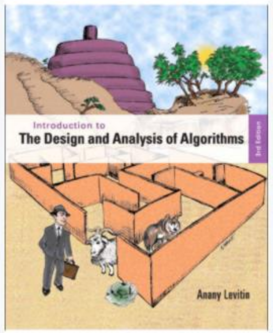
Reference Textbooks - Data Structures & Algorithms

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/blob/master/Reference-Textbooks

Data Structures and Algorithms in C++, Michael Goodrich	Data Structures and Algorithms Annotated Reference, Granville Barnett	Data Structures and Program Design in C++, Robert Kruse
		
Data Structures and Algorithms Made Easy, Narasimha Karumanchi	Data Structures and Algorithms in Python, Michael Goodrich	Data Structure and Algorithmic Thinking with Python, Narasimha Karumanchi
		

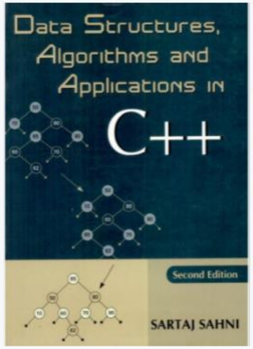
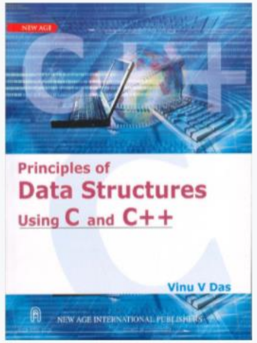
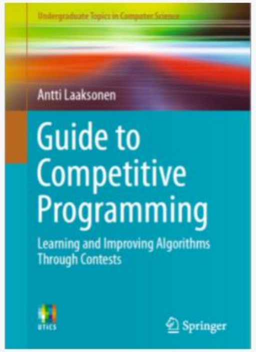
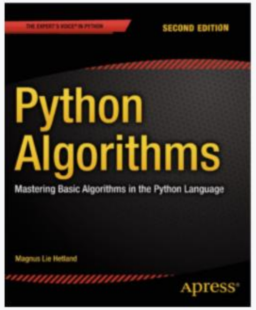
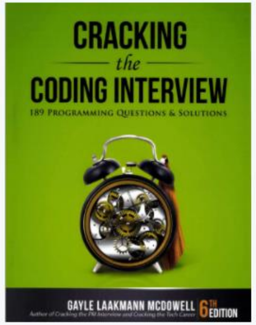
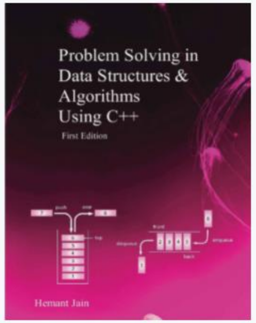
Reference Textbooks - Data Structures & Algorithms

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/blob/master/Reference-Textbooks

Introduction to Algorithms, Thomas H. Cormen	Analysis of Algorithms An Active Learning Approach, Jeffrey J. McConnell	Competitive Programming 3, Steven Halim
		
Fundamental of Algorithmics, Gilles Brassard and Paul Bartley	Grokking Algorithms, Aditya Y. Bhargava	Introduction to the Design and Analysis of Algorithms, Anany V. Levitin
		

Reference Textbooks - Data Structures & Algorithms

Repository Link: github.com/cs-MohamedAyman/Hands-On-Experience/blob/master/Reference-Textbooks

Data Structures and Algorithms and Applications in C++, Sartaj Sahni	Principles of Data Structures using C and C++, Vinu Das	Guide to Competitive Programming, Antti Laaksonen
		
Python Algorithms Mastering Basic Algorithms in the Python Language, Magnus Hetland	Cracking the Coding Interview, Gayle McDowell	Problem Solving in Data Structures and Algorithms Using C++, Hemant Jain
		

Lecture Agenda

1- Why this Training is Important

2- Training Content and Educational Projects

3- How to Practice on Online Competitions

4- Technical Tutorials and Reference Textbooks

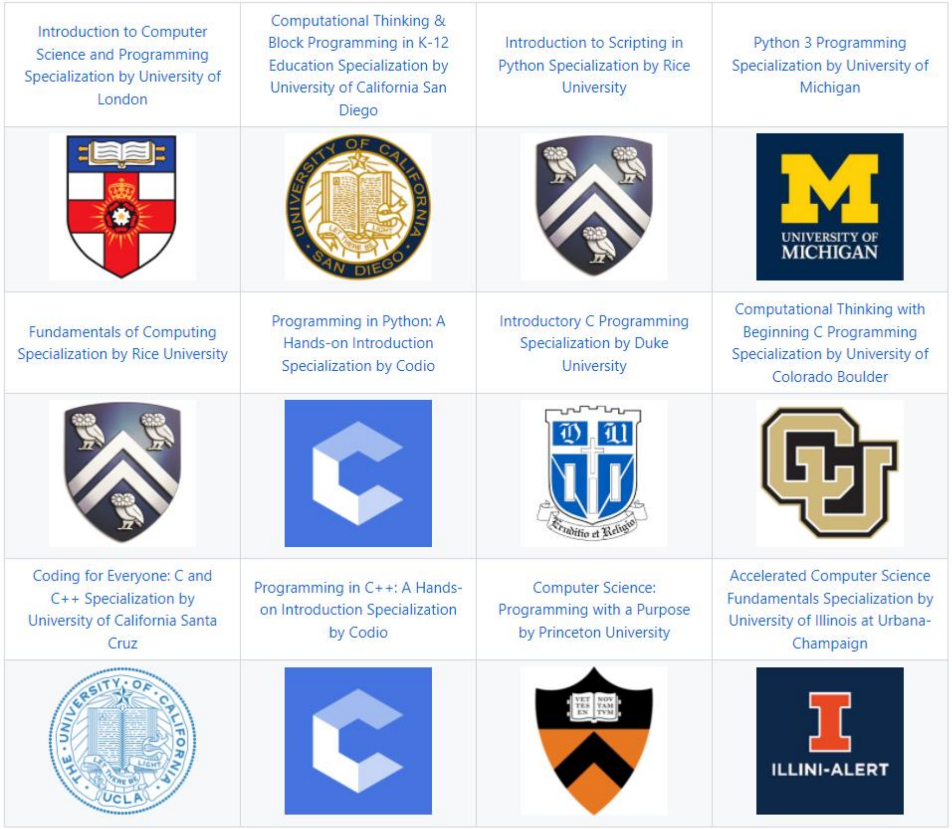
5- Online Learning Platforms Courses

6- Environment Setup and Installations

Coursera Specializations

Coursera - Programming Languages & Object-Oriented

Repository Link: github.com/cs-MohamedAyman/eLearning-Platforms/tree/master/Coursera-Specializations



Introduction to Computer Science and Programming Specialization by University of London	Computational Thinking & Block Programming in K-12 Education Specialization by University of California San Diego	Introduction to Scripting in Python Specialization by Rice University	Python 3 Programming Specialization by University of Michigan
			UNIVERSITY OF MICHIGAN
Fundamentals of Computing Specialization by Rice University	Programming in Python: A Hands-on Introduction Specialization by Codio	Introductory C Programming Specialization by Duke University	Computational Thinking with Beginning C Programming Specialization by University of Colorado Boulder
Coding for Everyone: C and C++ Specialization by University of California Santa Cruz	Programming in C++: A Hands-on Introduction Specialization by Codio	Computer Science: Programming with a Purpose by Princeton University	Accelerated Computer Science Fundamentals Specialization by University of Illinois at Urbana-Champaign
			ILLINI-ALERT

Coursera - Data Structures & Algorithms

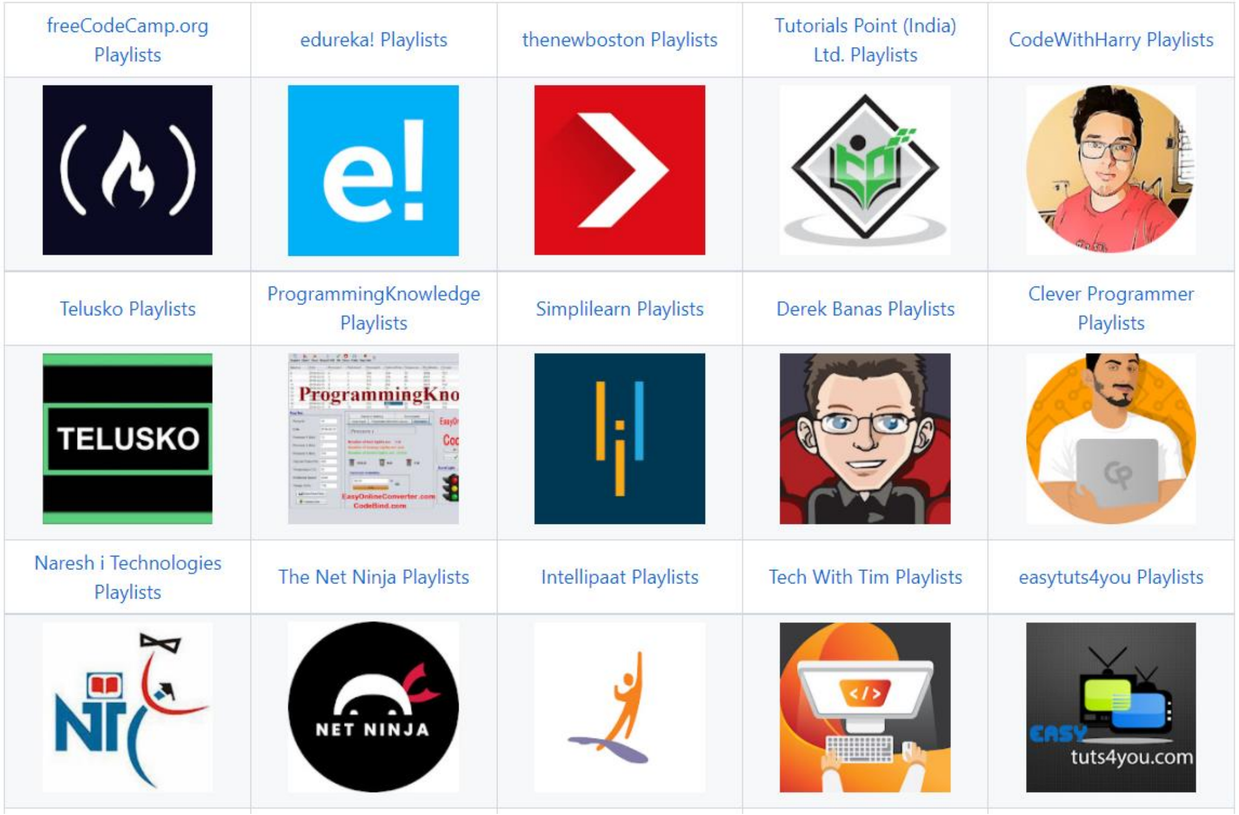
Repository Link: github.com/cs-MohamedAyman/eLearning-Platforms/tree/master/Coursera-Specializations

Introduction to Discrete Mathematics for Computer Science Specialization by University of California San Diego	Discrete Mathematics by Shanghai Jiao Tong University	Accelerated Computer Science Fundamentals Specialization by University of Illinois at Urbana-Champaign	Data Structures and Performance by University of California San Diego
Competitive Programming by Moscow Institute of Physics and Technology	Algorithms Specialization by Stanford University	Data Structures and Algorithms Specialization by University of California San Diego	Data Structures and Algorithms Specialization by University of Colorado Boulder
Algorithms by Princeton University	Geometric Algorithms by EIT Digital	Computational Geometry by Saint Petersburg State University	Computer Science: Algorithms, Theory, and Machines by Princeton University

YouTube Playlists

YouTube - Programming Languages & Object-Oriented

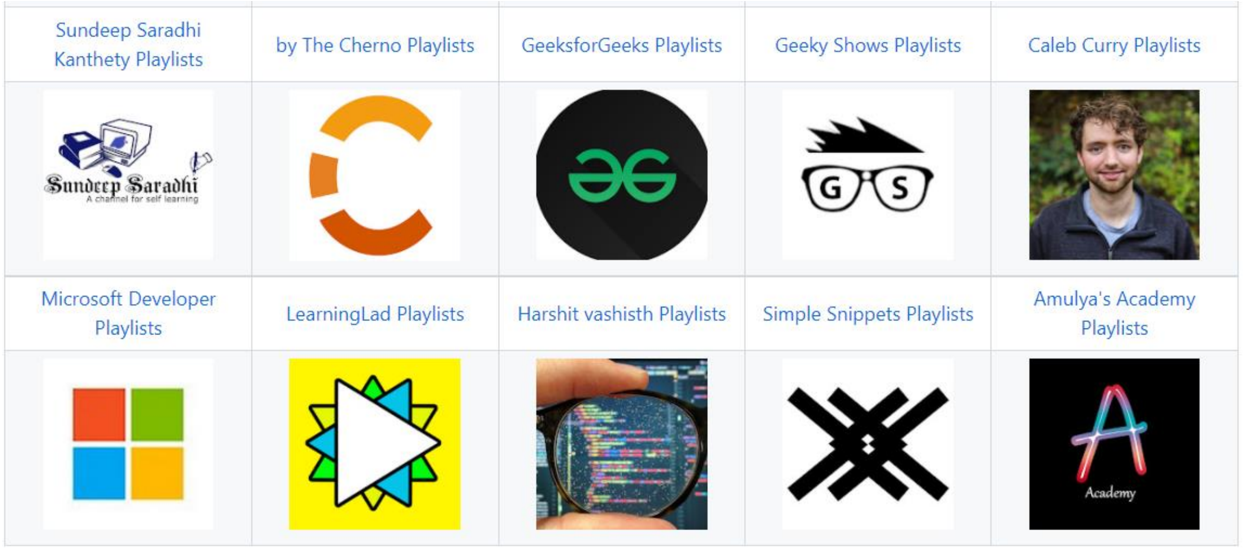
Repository Link: github.com/cs-MohamedAyman/eLearning-Platforms/tree/master/YouTube-Playlists



freeCodeCamp.org Playlists	edureka! Playlists	thenewboston Playlists	Tutorials Point (India) Ltd. Playlists	CodeWithHarry Playlists
Telusko Playlists	ProgrammingKnowledge Playlists	Simplilearn Playlists	Derek Banas Playlists	Clever Programmer Playlists
Naresh i Technologies Playlists	The Net Ninja Playlists	Intellipaat Playlists	Tech With Tim Playlists	easytuts4you Playlists

YouTube - Programming Languages & Object-Oriented

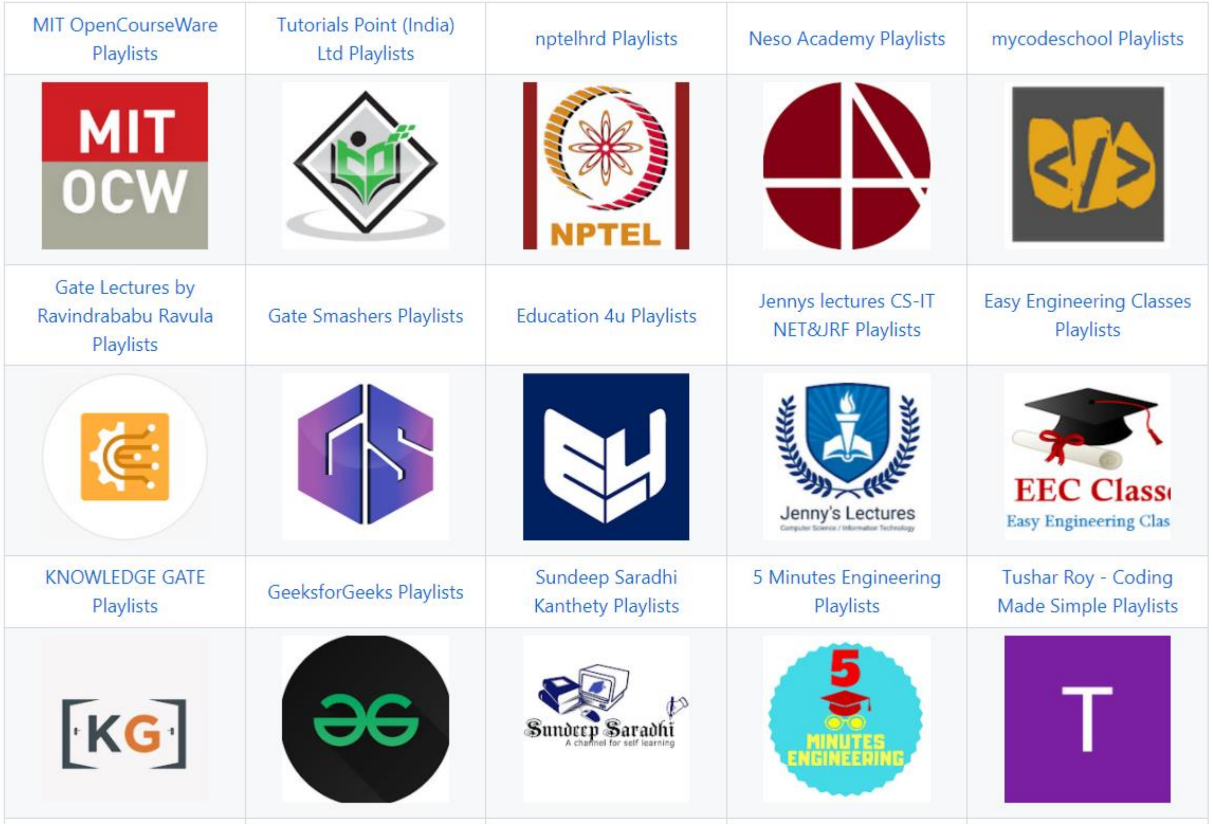
Repository Link: github.com/cs-MohamedAyman/eLearning-Platforms/tree/master/YouTube-Playlists



Sundeep Saradhi Kanthety Playlists	by The Cherno Playlists	GeeksforGeeks Playlists	Geeky Shows Playlists	Caleb Curry Playlists
Microsoft Developer Playlists	LearningLad Playlists	Harshit vashisth Playlists	Simple Snippets Playlists	Amulya's Academy Playlists

YouTube - Data Structures & Algorithms

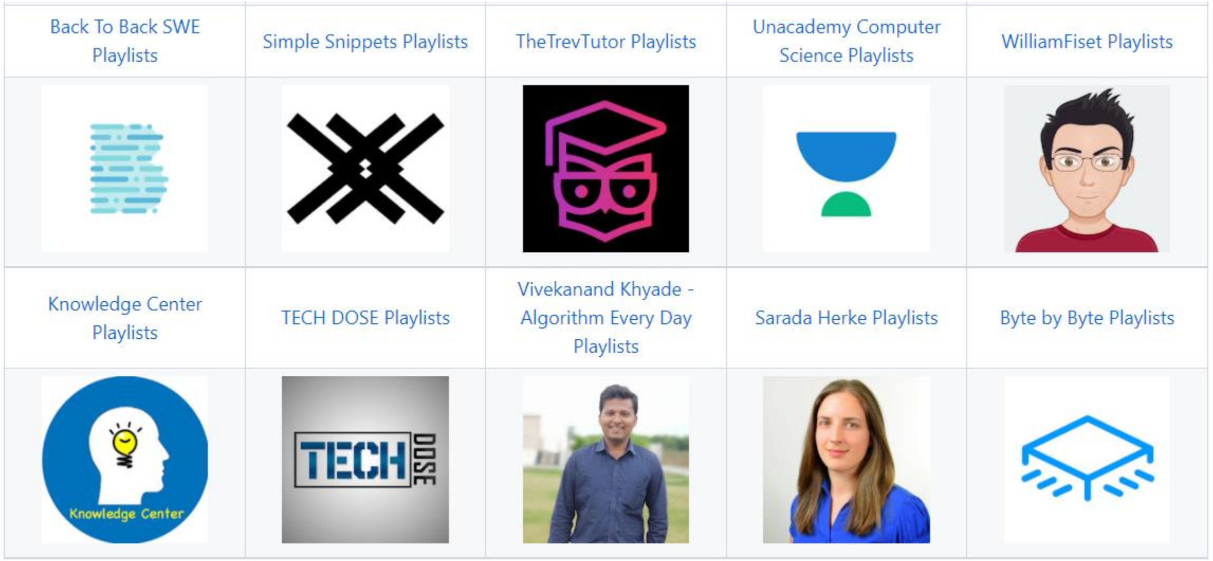
Repository Link: github.com/cs-MohamedAyman/eLearning-Platforms/tree/master/YouTube-Playlists



MIT OpenCourseWare Playlists	Tutorials Point (India) Ltd Playlists	nptelhrd Playlists	Neso Academy Playlists	mycodeschool Playlists
Gate Lectures by Ravindrababu Ravula Playlists	Gate Smashers Playlists	Education 4u Playlists	Jennys lectures CS-IT NET&JRF Playlists	Easy Engineering Classes Playlists
KNOWLEDGE GATE Playlists	GeeksforGeeks Playlists	Sundeep Saradhi Kanthety Playlists	5 Minutes Engineering Playlists	Tushar Roy - Coding Made Simple Playlists

YouTube - Data Structures & Algorithms

Repository Link: github.com/cs-MohamedAyman/eLearning-Platforms/tree/master/YouTube-Playlists



Back To Back SWE Playlists	Simple Snippets Playlists	TheTrevTutor Playlists	Unacademy Computer Science Playlists	WilliamFiset Playlists
Knowledge Center Playlists	TECH DOSE Playlists	Vivekanand Khyade - Algorithm Every Day Playlists	Sarada Herke Playlists	Byte by Byte Playlists

Lecture Agenda

1- Why this Training is Important

2- Training Content and Educational Projects

3- How to Practice on Online Competitions

4- Technical Tutorials and Reference Textbooks

5- Online Learning Platforms Courses

6- Environment Setup and Installations

Python & CPP
Online Environment

Python Online Interpreters

codechef.com/ide

replit.com/languages/python3

programiz.com/python-programming/online-compiler

onecompiler.com/python

tutorialspoint.com/execute_python3_online.php

geekflare.com/online-compiler/python

ideone.com

mycompiler.io/new/python

onlinegdb.com/online_python_interpreter

CPP Online Compilers

codechef.com/ide

replit.com/languages/python3

programiz.com/python-programming/online-compiler

onecompiler.com/python

tutorialspoint.com/execute_python3_online.php

geekflare.com/online-compiler/python

ideone.com

mycompiler.io/new/python

onlinegdb.com/online_python_interpreter

Python & CPP IDEs

Python Interpreters Installation

python.org

anaconda.com/distribution

jetbrains.com/pycharm

spyder-ide.org

CPP Compilers Installation

codeblocks.org

bloodshed.net

eclipse.org

visualstudio.microsoft.com

Lecture Agenda

1- Why this Training is Important

2- Training Content and Educational Projects

3- How to Practice on Online Competitions

4- Technical Tutorials and Reference Textbooks

5- Online Learning Platforms Courses

6- Environment Setup and Installations

DO
MORE.