May 1 – June 30 Detailed Study Plan (Monday–Thursday, 6:30–7:30 AM)

Date	Day	Topic (with detailed description)	Homework (small daily)	Thursday Homework (Big Project)
May 1	Thu	Python Basics (syntax, variables, comments, input/output, basic operations)	Solve 5 simple Python problems (calculator, area of circle, swap two numbers)	Build a basic calculator that can add, subtract, multiply, divide two numbers.
May 5	Mon	Python Flow Control (if-else conditions, for loops, while loops, nested loops)	Write 5 programs using different loops and conditions.	
May 6	Tue	Functions in Python (defining functions, parameters, return values, default parameters)	Create 5 functions like factorial, Fibonacci, even-odd checker.	
May 7	Wed	Python Lists (list creation, indexing, slicing, methods like append, remove, sort)	Solve 5 questions on lists like finding max, sum of elements, reversing a list.	
May 8	Thu	Tuples, Sets, Dictionaries (differences, operations, accessing elements)	Mini exercises on set operations and dictionary key-value access.	Create an address book program using a dictionary to store name, phone number.
May 12	Mon	File Handling (open, read, write, append files, file closing, context manager 'with')	Write a script to read a text file and print its contents.	
May 13	Tue	Error Handling (try, except blocks, multiple excepts, finally, raising errors)	Solve error-handling exercises, create 3 custom error cases.	

May 14	Wed	Modules and Libraries (importing libraries, using math, random, datetime)	Write small programs using random number generation, date/time formatting.	
May 15	Thu	SQL Basics (Database intro, SELECT, WHERE, INSERT)	Write 5 SELECT queries for filtering data.	Create a database table (students) and write queries to SELECT, INSERT data.
May 19	Mon	SQL Advanced Queries (GROUP BY, ORDER BY, HAVING, aggregate functions like SUM, AVG)	Write 5 queries using GROUP BY and ORDER BY.	
May 20	Tue	SQL JOINs (INNER JOIN, LEFT JOIN, RIGHT JOIN, JOIN syntax, examples)	Practice writing 5 JOIN queries between two tables (students and courses).	
May 21	Wed	SQL Subqueries and Views (nested queries, creating views)	Write 3 subqueries for selecting max salary, top student, etc.	
May 22	Thu	SQL Practice Session (mixing joins, subqueries, group by)	Solve complex queries combining JOIN and aggregation.	Design a database for a school (students, courses, teachers) with relationships.
May 26	Mon	Arrays (basic operations, inserting, deleting, traversing elements)	Solve 5 basic array problems.	
May 27	Tue	Linked List Basics (singly linked list creation, insertion, deletion)	Write Python code for a basic singly linked list.	
May 28	Wed	Stack and Queue (stack push/pop, queue enqueue/dequeue, real-life examples)	Implement stack in Python using list.	

May 29	Thu	Tree Basics (binary trees, nodes, traversals: pre-order, in-order, post-order)	Draw trees and perform simple traversals manually.	Build a basic Binary Tree class in Python and traverse it.
June 2	Mon	OOP Basics (Class, Object, Constructor, **init**(), methods)	Create a simple class with methods (like Car, Student).	
June 3	Tue	Inheritance and Polymorphism (parent-child classes, overriding methods)	Extend a class (e.g., Animal → Dog) with inheritance.	
June 4	Wed	Encapsulation and Abstraction (private/public variables, hiding details)	Write code using private variables and getter/setter methods.	
June 5	Thu	OOP Practice (combine all concepts: class, inheritance, polymorphism)	Solve 5 OOP structure exercises.	Create a Library Management System with book and member classes.
June 9	Mon	Python Interview Questions (variables, functions, classes, error handling)	Write answers to 10 common Python questions.	
June 10	Tue	SQL Interview Questions (joins, group by, aggregate queries, subqueries)	Practice answering 5 SQL conceptual questions.	
June 11	Wed	Data Structures Interview Questions (arrays, linked list, stack, queue)	Solve 5 DSA theory/coding questions.	
June 12	Thu	OOP Interview Questions (class vs object, types of inheritance, polymorphism examples)	Solve 5 theory questions and code 2 OOP problems.	Build an Employee Management System using OOP principles.

June 16	Mon	Python Coding Challenges (leetcode easy questions: strings, arrays)	Solve 3 easy problems (ex: two sum, palindrome checker).	
June 17	Tue	SQL Query Practice (more complex joins, window functions basics)	Solve 5 SQL query problems from HackerRank.	
June 18	Wed	DSA Problems (binary trees, linked list reversal)	Solve 3 problems: tree traversal, reverse a linked list, stack operations.	
June 19	Thu	Mock Interview Practice (simulate coding interview)	Attempt solving 5 questions with time limit (20 mins each).	Write a full mock interview question set with SQL + Python + DSA.
June 23	Mon	Final Revision: Python + SQL (all quick notes, syntax revision)	Attempt small quizzes (MCQs).	
June 24	Tue	Final Revision: DSA + OOP (important concepts revision)	Quick short answer questions.	
June 25	Wed	Practice Final Test (Python, SQL, DSA all mixed)	Attempt a full practice paper.	
June 26	Thu	Certificate Exam Preparation (register, prepare documents)	Fill exam application form.	Apply for exam and review final notes.
June 30	Mon	Take Certification Exam (or final preparation if exam delayed)		

Certificates Suggested:

- Google Python Certificate

- FreeCodeCamp Data Analysis with Python
- HackerRank SQL Certificate (free)
- Microsoft Python Certification (paid)