

## COURSE GLOSSARY

# Data Structures and Algorithms in Python

**Algorithm:** A finite sequence of well-defined instructions or steps designed to solve a specific problem or perform a computation

**Big O notation:** A mathematical notation used to describe the upper bound (worst-case) asymptotic growth rate of an algorithm's time or space requirements as input size grows

**Binary search tree (BST):** A binary tree with the ordering property that every node's left subtree contains values less than the node and the right subtree contains values greater than the node, enabling efficient search, insertion, and deletion

**Binary tree:** A hierarchical tree data structure where each node has at most two children commonly referred to as the left and right child

**Call stack:** The runtime stack structure that records active function calls, their local variables, and return addresses, which grows and shrinks as recursive or nested calls are made and resolved

**Collision:** A situation in a hash table where two distinct keys produce the same hash value and map to the same slot, requiring a collision-resolution strategy

**Data structure:** A way of organizing and storing data in memory to enable efficient access and modification, such as arrays, linked lists, trees, and graphs

**Hallucination:** When a model produces confident but incorrect or fabricated information, often due to gaps or biases in its training data or reasoning process

**Hallucination:** When a model produces confident but incorrect or fabricated information, often due to gaps or biases in its training data or reasoning process

**Graph:** A collection of vertices (nodes) connected by edges (links) that can be directed or undirected and optionally weighted, used to model pairwise relationships

**Hash function:** A deterministic function that maps keys to integer indices (slots) in a hash table, producing the same output every time for the same input

**Hash table:** A data structure that stores key-value pairs by applying a hash function to keys to determine the slot or bucket where each value is stored, enabling fast lookups

**Linked list:** A linear data structure made of nodes where each node stores data and a reference to the next node, allowing elements to be stored in non-contiguous memory

**Node:** A single element of a node-based data structure that typically contains a data field and one or more references (pointers) to other nodes

**Pointer:** A reference stored in a node that indicates the location of another node or points to None (null) when there is no adjacent node

**Pop:** The operation that removes and returns the top element from a stack

**Push:** The operation that adds (inserts) an element onto the top of a stack

**Queue:** A collection that follows the First-In-First-Out (FIFO) principle where elements are removed in the same order they were added

**Recursion:** A programming technique in which a function calls itself directly or indirectly to solve a problem by breaking it into smaller subproblems, requiring a base case to terminate

**Singly linked list:** A linked list in which each node contains exactly one pointer referencing the next node, forming a one-directional chain from head to tail

**Space complexity:** A measure, expressed asymptotically, of the extra memory an algorithm requires (beyond the input itself) as a function of input size

**Stack:** A collection that follows the Last-In-First-Out (LIFO) principle where the most recently added element is the first removed

**Time complexity:** A measure, expressed asymptotically (often with Big O), of how the runtime of an algorithm increases with the size of its input