



Stacks

Stacks

- Last in First Out (LIFO)
 - Processing Data in Reverse Order
 - Time
 - Backtracking
 - Call Stack
- Stack Operations
 - Push: Add new element to top of the stack
 - Pop: Remove and return top element of the stack
 - Peek: Observe but not remove the first element in a Stack
 - Print: Print all elements in a Stack

Stack



Stacks

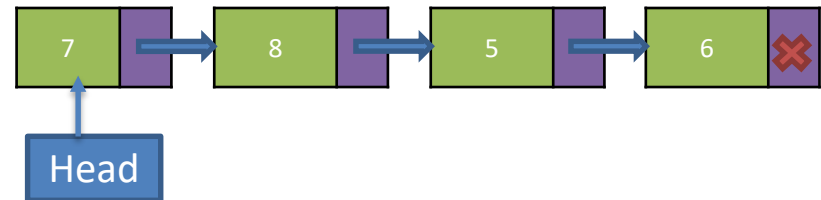
- Stack Implementations
 - Array
 - Linked List
- One Major Reference
 - Head

Array Stack

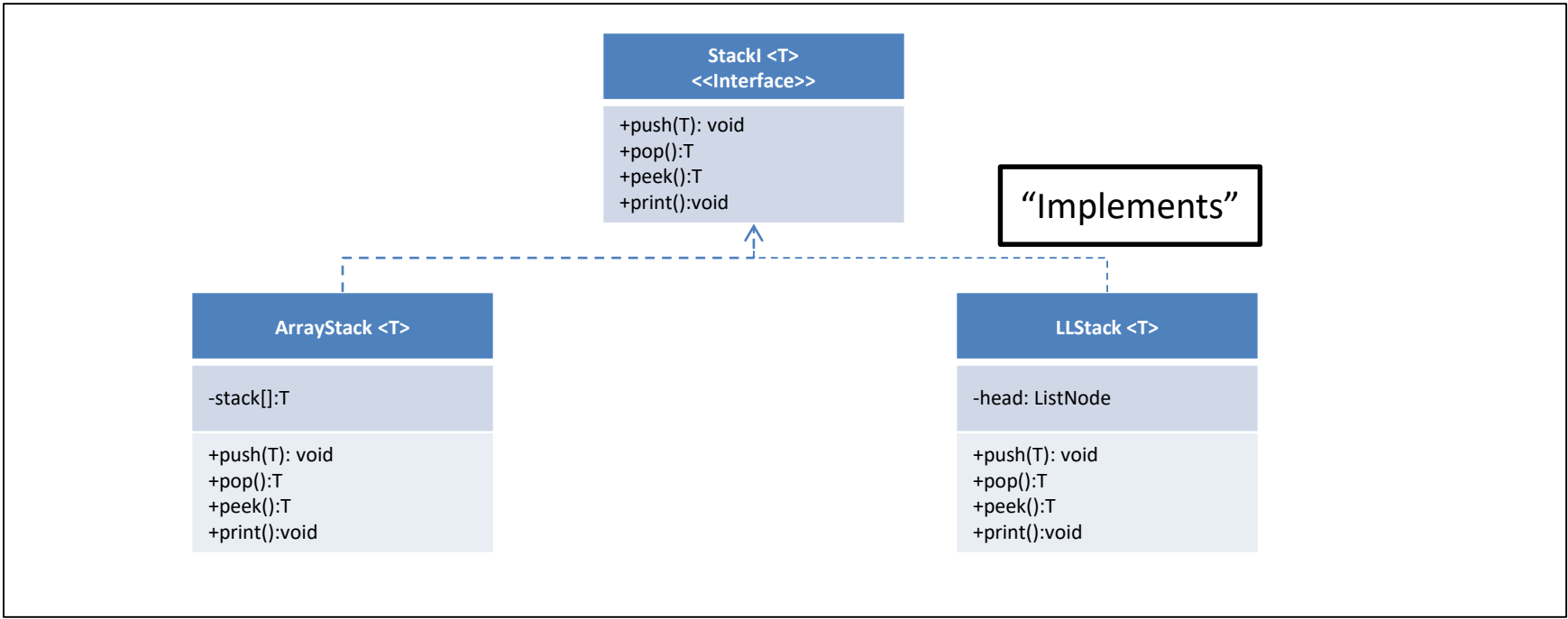
Index	0	1	2	3	4
Value	6	5	8	7	-

↑
Head Index

Linked List Stack



Stacks



Array Stack

- References
 - Head Index (First NULL Element)
 - Items in the Stack start from the Head Index - 1 and goes until index 0

Array Stack

Index	0	1	2	3	4	5
Value	7	8	5	6	-	-

↑
Head Index

Array Stack

- Push
 - Add element at the Head Index
 - Increase the Head Index by 1

Array Stack

Index	0	1	2	3	4	5
Value	7	8	5	6	-	-

Head Index

Array Stack

- Push
 - Add element at the Head Index
 - Increase the Head Index by 1

Array Stack

Index	0	1	2	3	4	5
Value	7	8	5	6	4	-

Head Index

Array Stack

- Push
 - Add element at the Head Index
 - Increase the Head Index by 1

Array Stack

Index	0	1	2	3	4	5
Value	7	8	5	6	4	-

Head Index

Array Stack

- Pop
 - Save a reference to the data stored in Head Index – 1
 - Move the Head Index Backwards
 - Return the stored data

Array Stack

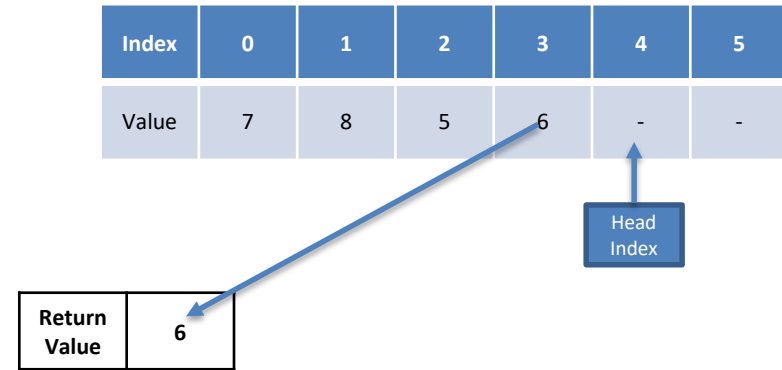
Index	0	1	2	3	4	5
Value	7	8	5	6	-	-

Head Index

Array Stack

- Pop
 - Save a reference to the data stored in Head Index – 1
 - Move the Head Index Backwards
 - Return the stored data

Array Stack



Array Stack

- Pop
 - Save a reference to the data stored in Head Index – 1
 - Move the Head Index Backwards
 - Return the stored data

Array Stack

Index	0	1	2	3	4	5
Value	7	8	5	6	-	-

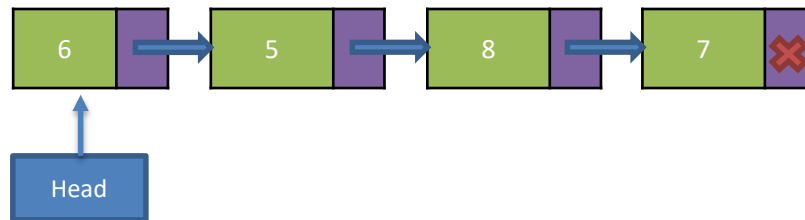
Head Index

Return Value	6
--------------	---

Linked List Stack

- Push
 - Create a new List Node with the Data
 - Point new List Node's reference to Head
 - Update the Head Reference to the new List Node

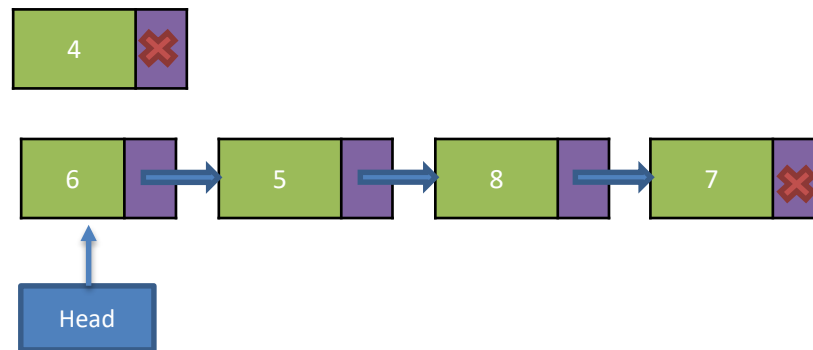
Linked List Stack



Linked List Stack

- Push
 - Create a new List Node with the Data
 - Point new List Node's reference to Head
 - Update the Head Reference to the new List Node

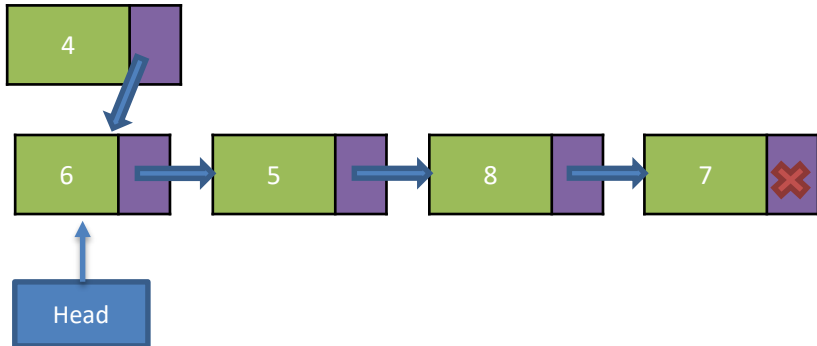
Linked List Stack



Linked List Stack

- Push
 - Create a new List Node with the Data
 - Point new List Node's reference to Head
 - Update the Head Reference to the new List Node

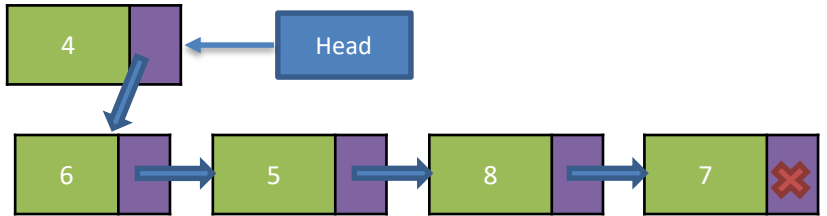
Linked List Stack



Linked List Stack

- Push
 - Create a new List Node with the Data
 - Point new List Node's reference to Head
 - Update the Head Reference to the new List Node

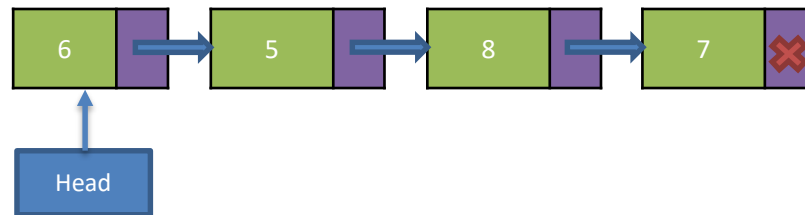
Linked List Stack



Linked List Stack

- Pop
 - Save reference to data store in the head
 - Move Head forward / Head = Head.link
 - Return the stored data

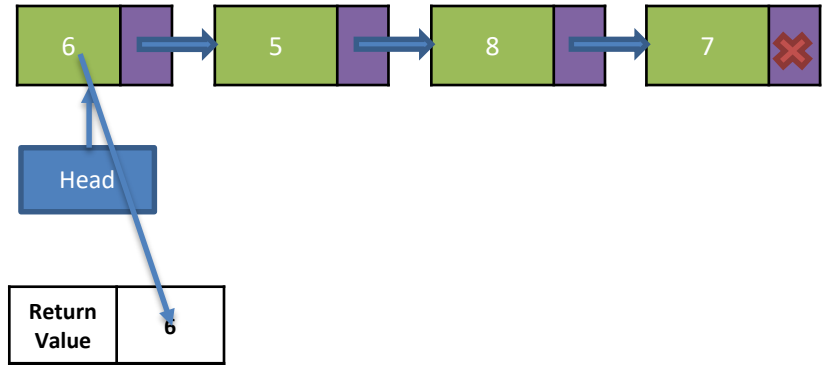
Linked List Stack



Linked List Stack

- Pop
 - Save reference to data store in the head
 - Move Head forward / Head = Head.link
 - Return the stored data

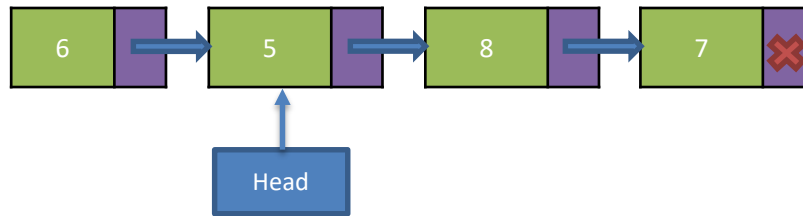
Linked List Stack



Linked List Stack

- Pop
 - Save reference to data store in the head
 - Move Head forward / Head = Head.link
 - Return the stored data

Linked List Stack

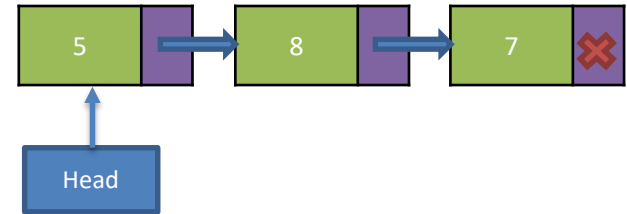


Return Value	6
--------------	---

Linked List Stack

- Pop
 - Save reference to data store in the head
 - Move Head forward / Head = Head.link
 - Return the stored data

Linked List Stack



Return Value	6
--------------	---