Stacks

- Ordered collection data, with two principal operations
  - Push: adds an element to the top of the collection
  - Pop: removes the last element from the collection
- LIFO (Last In, First Out)
Stacks

Operations

- Initialize Stack
- Push
- Pop
- Status
  - isEmpty(), isFull(), getItemSize() (dissertation)
- Clear Stack
Implementations

- Arrays
  - Fixed Size

- Linked List
  - Dynamically increase/shrink stack size

Array representation

- Two variables are needed
  - Array of the data type
  - For a static array, a current index of the top of the stack
  - For a dynamic array, a size of the stack
Implementation with an Array

- Check “Stacks.cpp”
- Check “Stacks_Class.cpp”

Implementation with a Linked List

- The header of the linked list points to the top of the stack

Base pointer

- Push(): inserts an element to the front of the list
- Pop(): removes an element from the front of the list
Implementation with a Linked List

- Check “Stacks_LinkedList.cpp”

Examples of Stacks

- Evaluation of Arithmetic Expression
- Infix to Postfix Conversion (HW2)
- Function parameter
Examples 1

- Evaluation of Arithmetic Expression
  - Infix and Postfix Expression
    - Infix expression
      - precedence and associativity of operators
      - Parenthesis
    - Postfix expression
      - No Parentheses
  - Examples:
    - Infix: $4 \times (2 + 4) - 3$
    - Postfix: $424+*3-$

Evaluation of Arithmetic Expression

- Rule for postfix expression
  - If an operand: push it on the stack
  - If an operator: pop 2 operands and perform the computation, and push the result

- Check “Postfix evaluation.cpp”
Function parameters

- Call a function with parameters: push
- In the function:
  - pop when the function is finished
  - push the result of the function
- Pop the result

Recursive function

- For example, factorial Function

```c
int factorial(int n)
{
    int r = 1;
    if (n == 1) return 1;
    r = n * factorial(n - 1);
    return r;
}
```

- x = factorial(2):
  - push(n=2) -> push(r=1) -> push(n=1) -> push(r=1) -> push() -> pop() -> pop() -> push(fac=1) -> ...