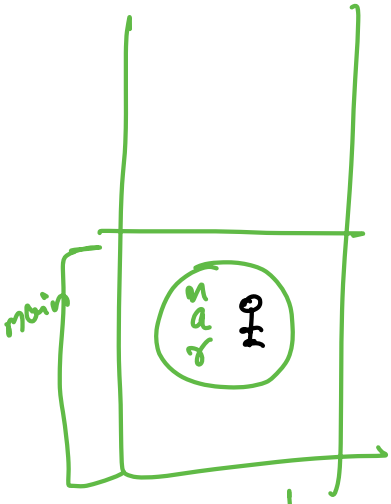
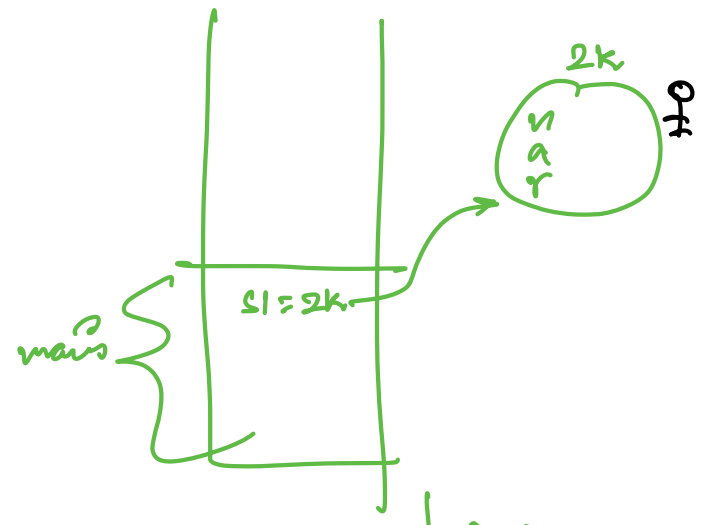


main
Student s;
↓
Stack

Student *s1 = new Student();
↓
Heap



s.n | (ks) → n
s.a | (ks) → a
s.r | (ks) → r



s1 → n | (*s1).n
s1 → a | (*s1).a
s1 → r | (*s1).r

inside the m-location?

address →

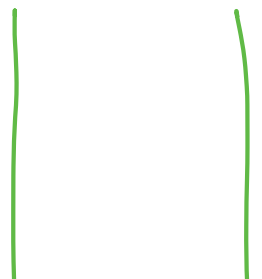
this

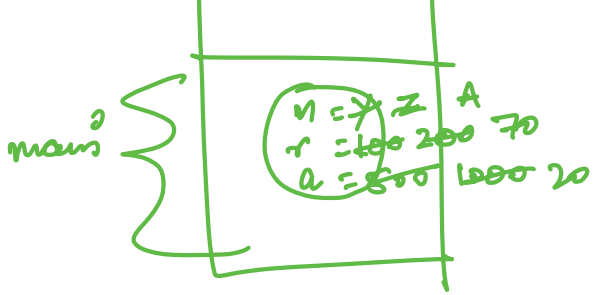
Constructor:

- frst name exactly class name
- no return type

Task: D.M initialization

- students / new student
1. memory allocate
 2. data area / parsing
 3. constructor code

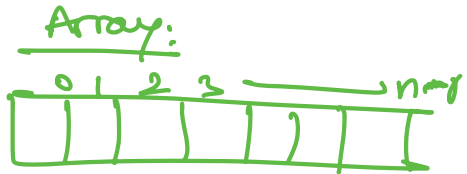
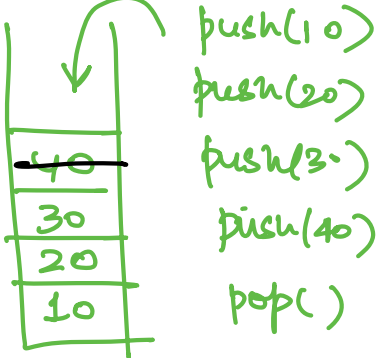




1. d.c odd compiler

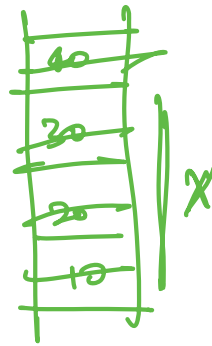
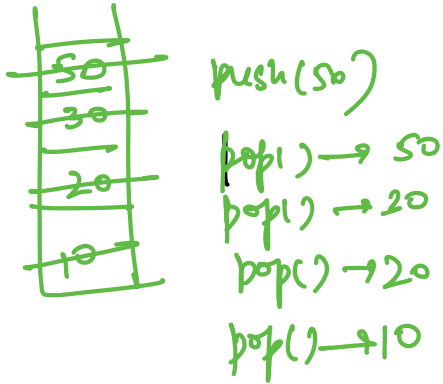
STACK

Data Structure
end



$O(1)$

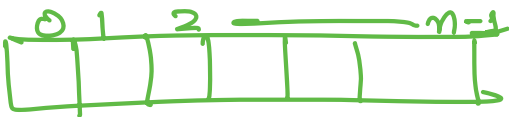
push, pop, top
 $O(1)$



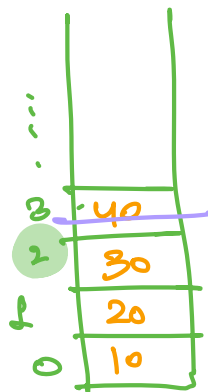
Implementation:

Stack → array ✓
 → linked list

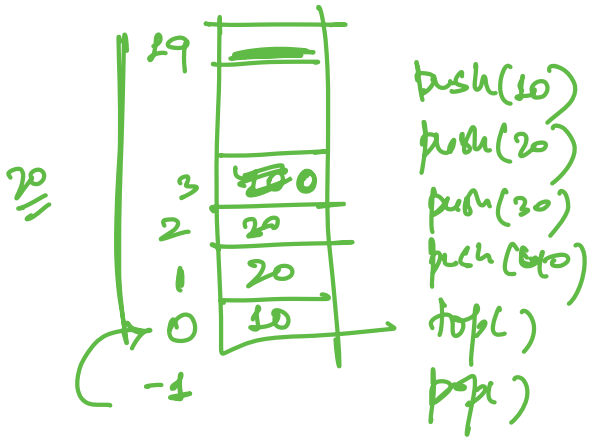
Internally: array



tos
(top of stack)
tos = 2



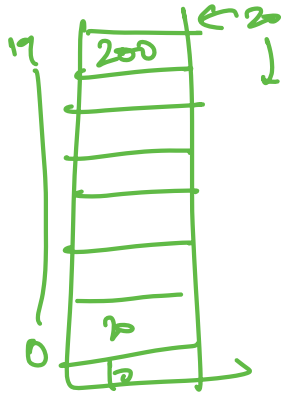
push(10)
 (20)
 (30)
 (40)
 tos = 2
 pop()
 tos = 2



```

while (top > 0)
{
  cout << arr[top];
  top--;
}
  
```

X tos will change.



Abstraction