

Stack vs Heap

Stack

```
#include <iostream>
using namespace std;
```

```
int add(int num1, int num2) {
    return num1 + num2;
}
```

```
int main() {
    int a = 2;
    int b = 3;
```

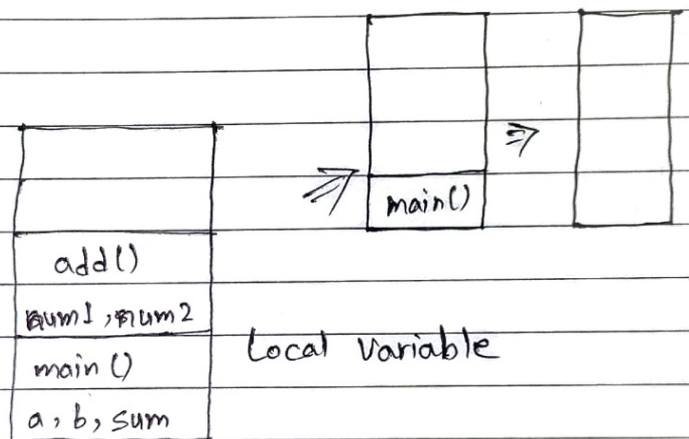
```
    int sum = add(a, b);
    cout << sum << endl;
```

```
    return 0;
```

```
}
```

Stack memory Allocation

temporary memory area uses
by program to store its local
variable



memory Stack

* The memory where local variables are stored temporarily is called heap.

* memory freed when function exits by compiler.

Teacher's Signature : _____

Stack

```
#include <iostream>
using namespace std;
```

```
void func1() { // function body }
void func2() { // function body }
void func3() { // function body }
void func4() { // function body }
```

```
int main() {
```

```
    func1();
```

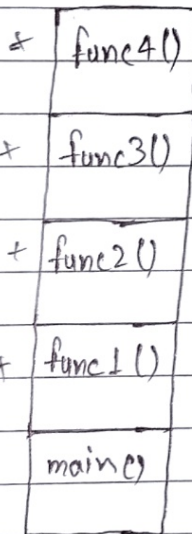
```
    func2();
```

```
    func3();
```

```
    func4();
```

```
    return 0;
```

```
}
```



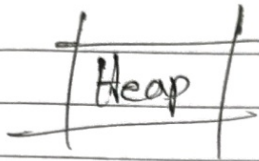
Stack Overflow

Memory Stack

* stack has fixed limited space hence it can run out of storage which causes stack overflow.

* stack overflow happens when run out of memory due to indefinitely call of a function in a function.

Teacher's Signature : _____



* size is not fixed

Heap (Dynamic)
memory
allocation

```
#include <iostream>
using namespace std;
```

```
int main() {
```

```
    int a = 10; // stored in stack
```

```
    int *p = new int(); // allocate memory in heap
```

```
    *p = 10;
```

```
    delete (p); // deallocate memory
```

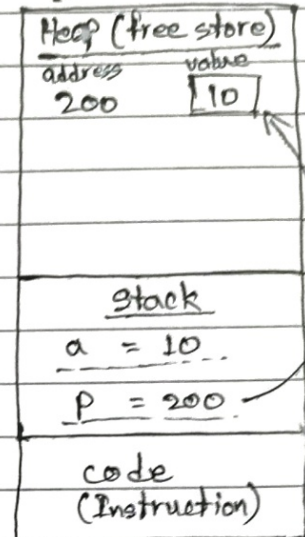
```
    p = new int[4]; // allocate memory for
                    // array in heap
```

```
    delete [] p; // deallocates p
```

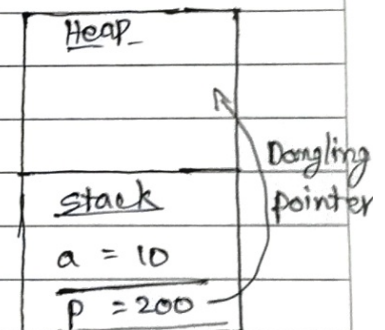
```
    p = NULL; // removes address stored in p
```

```
    return 0;
```

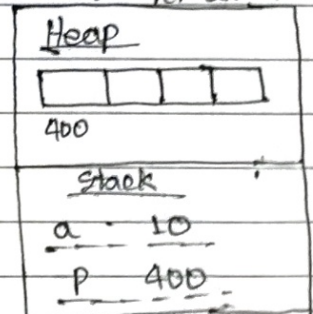
```
}
```



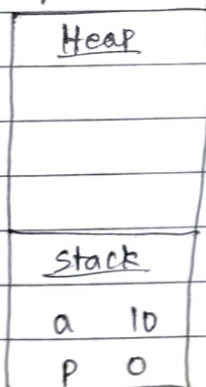
↓
Deallocation



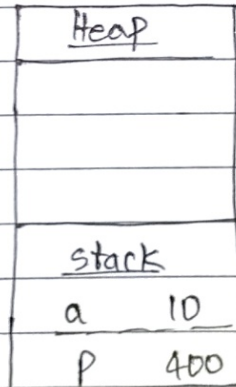
↓
allocation for array



p = NULL;



delete [] p;



Teacher's Signature : _____

Brief:

Stack : ① temporary memory area used to store program's local variable.

② It is fixed.

③ It is Last In First Out data structure (LIFO)

Stack overflow:

① It happens when program tries to access more memory than available stack size.

Heap : ① Permanent memory area used to store program's data.

② It is not fixed during compilation

Dangling pointer:

When a pointer points to freed or deallocated memory address is called Dangling pointer

Stack Memory allocation:

① Memory is allocated on the function call stack

② Memory gets deallocated when function call gets over

③ Deallocation handled by compiler

Heap memory allocation:

① Allocation takes place on the pile of memory space available to programmers to allocate and de-allocate.

② Programmer has to handle the deallocation.

Teacher's Signature : _____

Allocate memory in heap:

Syntax:

<data-type>* myPointer = new <data-type> [size]

int *p = new int [10];

Deallocation:

delete (p);

delete [] p;

Operators:

① delete operator:

delete (p); // deallocates p

delete [] p; // deallocates array

② New operator:

int *p = new int [10];

* Practice Questions:

1. Declare a 2D array Dynamically.
2. Declare a 3D array Dynamically.
3. MCQ on Dynamic Memory Allocation.

* To avoid dangling pointers use NULL to nullify the pointer.
pointer = NULL;

Teacher's Signature : _____