DMA (Dynamic Memory Alloc)

C

malloc ( )

free ( )

Both are operators as well as Keyword

Both are predefined fn

C++

new
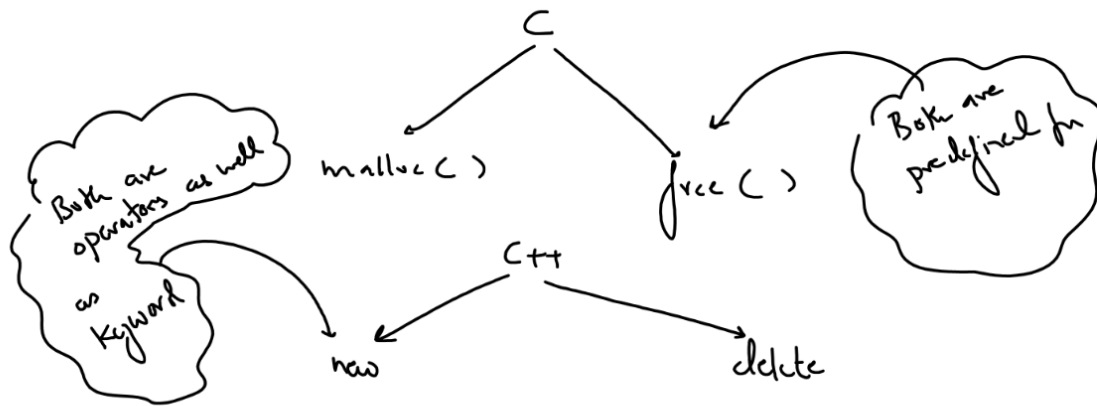
delete

Syntax of new

① new <data type> ;   *for allocating memory for single*

② new <data type> [size] ;   *No of elements*

Example

Allocating memory for 1 ele

①
```
int * p;

p = new    int ;

*p = 15;
```

heap
```
┌──────┐
│  + 15 │
└──────┘
5000    5004
```
```
┌──────┐
│ 5000 │
└──────┘
   p
```

Allocating memory for array

②
```
int *p;
p = new int [ 5 ];
```

can be replaced with a var

heap
```
┌──┬──┬──┬──┬──┐
│ ? │ ? │ ? │ ? │ ? │
└──┴──┴──┴──┴──┘
6000              6020
```
```
┌──────┐
│ 600  │
└──────┘
   p
```
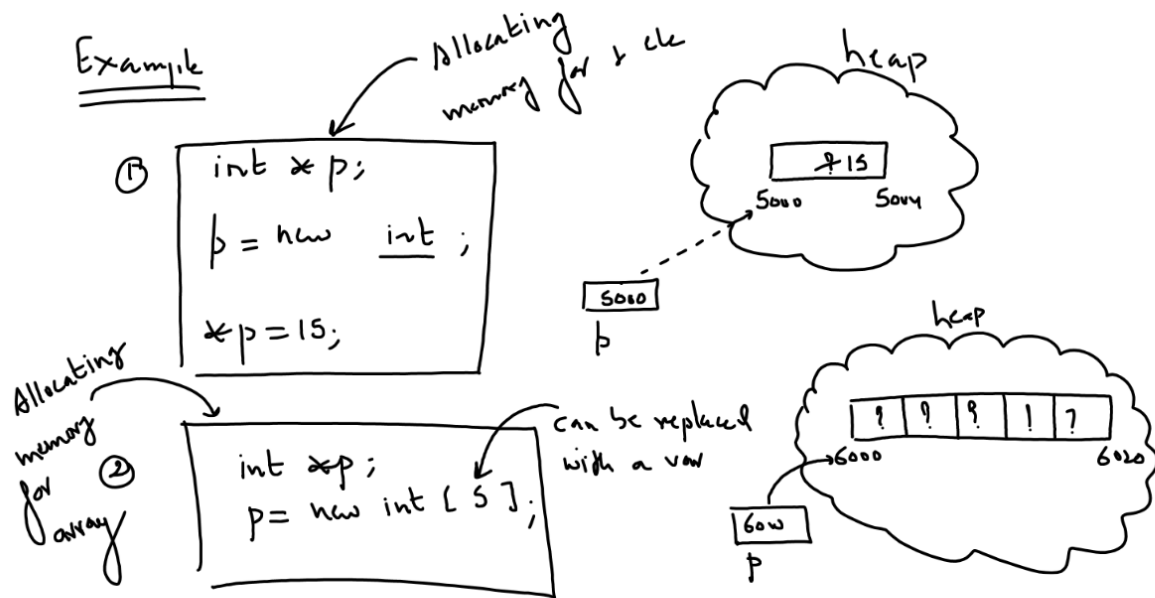
Syntax of delete

① delete <ptr-name>;

for deallocating single ele

② delete [ ] <ptr-name>;

for deallocating entire array

Example

①
```
int * p;

p = new    int ;

*p = 15;

delete p;
```
Allocating memory for 1 ele

5000   *15
        5

5010
p

② Allocating memory for array
```
int *p;
p = new int [ 5 ];

delete []p; ✓
```
Can be replaced with a var

? ?
6000 6004

600
p
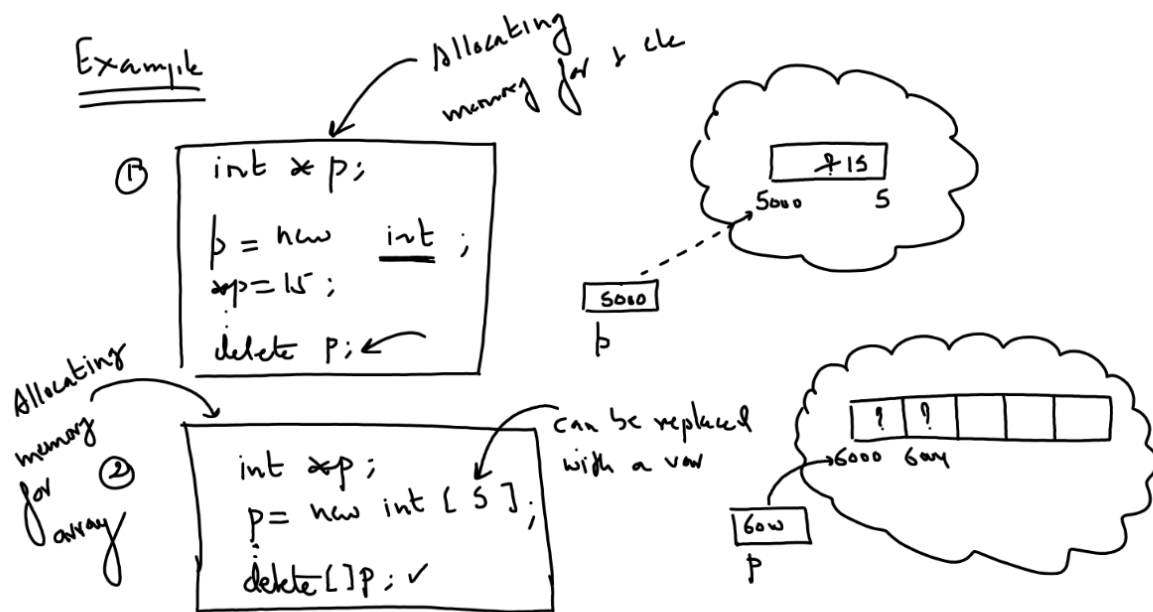
WAP to create a class called MyArray having following data members:

1. An integer pointer

2. An integer n for storing size of the array

3. An integer called sum for storing sum of all the elements of the array.

Also provide following member functions:

1. A constructor for accepting size of the array from the user and creating a dynamic array of that size.

2. A member function called get() for inputting values in the dynamic array.
3. A member function called calculate() for adding all the elements of the dynamic array.
4. A member function called show() for displaying all the elements as well as their sum.
5. An appropriately coded destructor

```cpp
#include <iostream>
#include <stdlib.h>
using namespace std;
class MyArray
{
    int *p;
    int n;
    int sum;
public:
    MyArray();
    void get();
    void calculate();
    void show();
    ~MyArray();
};
MyArray::MyArray()
{
    cout<<"enter size:";
    cin>>n;
    p=new int[n];
    if(p==0)
    {
        cout<<"Memory
insufficient";
        exit(1);
    }
    sum=0;
```

```cpp
void MyArray::get()
{
    for(int i=0;i<n;i++)
    {
        cout<<"Enter value:";
        cin>>p[i];
    }
}
void MyArray::calculate()
{
    for(int i=0;i<n;i++)
    {
        sum+=p[i];
    }
}
void MyArray::show()
{
    cout<<"Values are:"<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<p[i]<<endl;
    }
    cout<<"Their sum is "<<sum;
}
MyArray::~MyArray()
{
    delete []p;
}
```

```cpp
int main()
{
    MyArray obj;
    obj.get();
    obj.calculate();
    obj.show();
    return 0;
}
```

Will terminate the code

obj

p	8w
n	7
s~	0

heap

8w