Virtual Constructor in C++

The virtual mechanism works only when we have a base class pointer to a derived class object.

In C++, the constructor cannot be virtual, because when a constructor of a class is executed there is no virtual table in the memory, means no virtual pointer defined yet. So, the constructor should always be non-virtual.

But virtual destructor is possible.

Example Code

```
#include<iostream>
using namespace std;
class b {
   public:
      b() {
         cout<<"Constructing base \n";
      }
      virtual ~b() {
         cout<<"Destructing base \n";
      }
};
class d: public b {
   public:
      d() {
         cout<<"Constructing derived \n";
      }
      ~d() {
         cout<<"Destructing derived \n";
      }
};
int main(void) {
   d *derived = new d();
   b *bptr = derived;
   delete bptr;
   return 0;
}
```

Output

```
Constructing base
Constructing derived
Destructing derived
Destructing base
```