NEW AND DELETE

Creating and destroying objects on the heap

Delroy A. Brinkerhoff

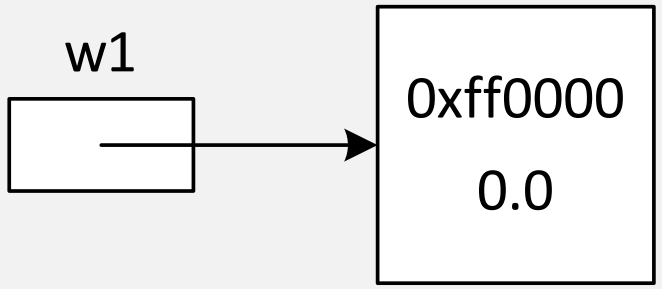
CLASS EXAMPLE

```
class widget
{
    private:
        int       color = 0xff0000;
        double    cost = 0.0;
    public:
        widget() {}
        widget(int a_color, double a_cost)
            : color(a_color), cost(a_cost) {}
        ~widget() { . . . }
        int get_color() { return color; }
};
```

DYNAMIC OBJECTS

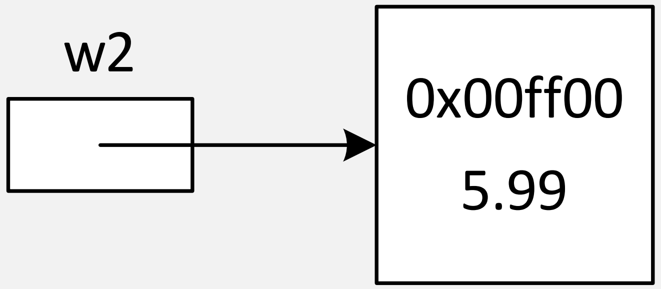
DEFAULT CONSTRUCTOR

- `widget* w1 = new widget;`
- `delete w1;`



GENERAL CONSTRUCTOR

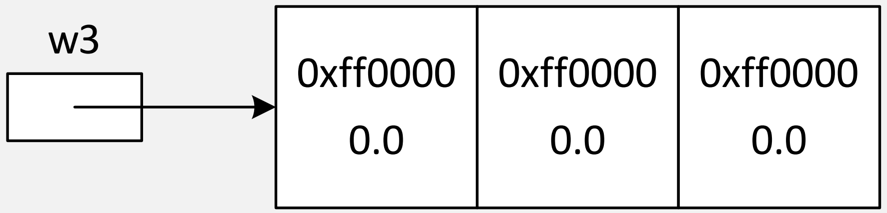
-
  ```
  widget* w2 =
     new widget(0x00ff00, 5.99);
  ```
- `delete w2;`



ARRAYS AND OBJECTS

ARRAY OF OBJECTS

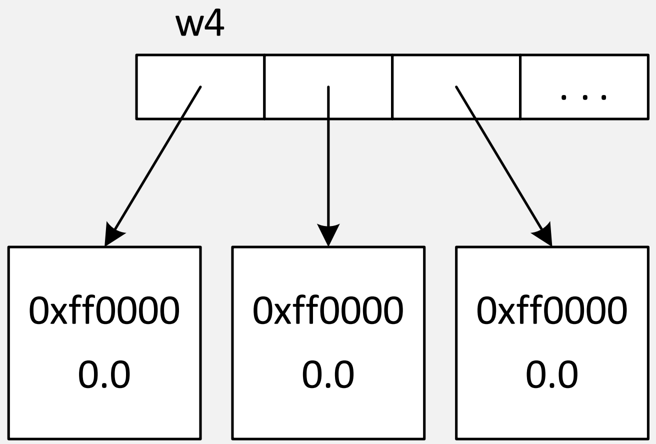
- `widget* w3 = new widget[n];`
- `delete[] w3;`



ARRAY OF OBJECT POINTERS

- `widget* w4[100];`

-
  ```
  for (int i = 0; i < 3; i++)
          w4[i] = new widget;
  ```



ARRAYS AND FEATURE ACCESS

ARRAYS OF OBJECTS

- `widget w5[100];`
- `w5[10].get_color()`

- `widget* w6 = new widget[100];`
- `w6[10].get_color()`

ARRAYS OF POINTERS

- `widget* w7[100];`
-
```
for (int i = 0; i < 100; i++)
    w7[i] = new widget(0x0000ff, 10.0);
```
- `w7[10]->get_color()`