

THE BIG FOUR

Problem Solving with Computers-II



Read the syllabus. Know what's required. Know how to get help.

CLICKERS OUT

How is h01 (specifically the CS16 final) going?

- A. Done - I think I have done well
- B. Attempted - found it a bit difficult
- C. Attempted - found some concepts alien
- D. Attempted - extremely difficult
- E. Haven't attempted

Clickers out – frequency AC

The Big Four

Special methods (member functions) of any C++ class.

1. Constructor → used for setup / initialization routines
2. Destructor → used for running "tear-down" routines
3. Copy Constructor → used to initialize a new object using an existing object
4. Copy Assignment → used to assign the values of one object to another (via the '=' operator)

This lecture we will talk about:

1. Why do the Big Four exist?
2. When is each method called?

$a x^2 + b x + c$ Quadratic expression

Define a class to represent any quadratic expression

```
class Quadratic {
```

```
    public:
```

```
        double evaluate(double x);
```

```
    private:
```

```
        double a;
```

```
        double b;
```

```
        double c;
```

```
};
```

We'll use this class as our running example to discuss the Big Four!

Constructor and Destructor

Every class has the following special methods:

- Constructor: Called right AFTER new objects are created in memory
- Destructor: Called right BEFORE an object is deleted from memory

The compiler automatically generates default versions, but you can override them

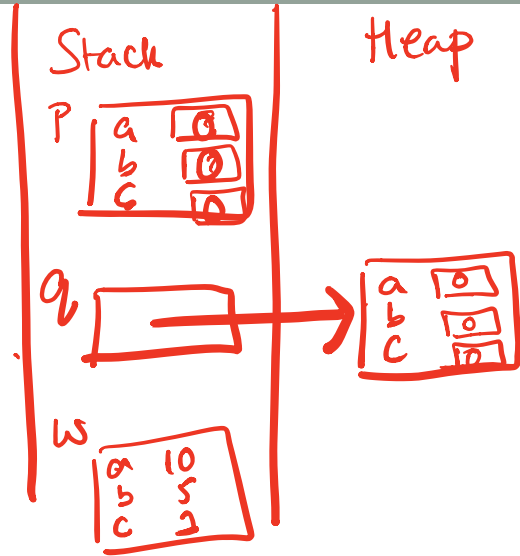
Constructor (last class)

```
void foo(){  
    Quadratic p; // default  
    Quadratic* q = new Quadratic;  
    Quadratic w(10, 5, 1);  
}
```

How many times is the constructor called in the above code?

- A. Never
- B. Once
- C. Twice
- D. ~~Thrice~~

Three times



We could also use the parameterized constructor to initialize an object on the heap

`Quadratic* w = new Quadratic(1, 2, 3);`

Initialization lists

- * Used to initialize member variables at the time they are created
- * Must be used to initialize constant member variables

```
Quadratic :: Quadratic (double aa, double bb, double cc)
: a(aa), b(bb), c(cc) // This line is the initialization list
{ }
Read a(aa) as a = aa;
```

Destructor

- Must have the same name as the class preceded by a ~ (tilda)
- Does not have a return type
- Called right BEFORE an object is deleted from memory

(Remember heap objects are only deleted using the keyword delete)

```
Quadratic::~~Quadratic() {
```

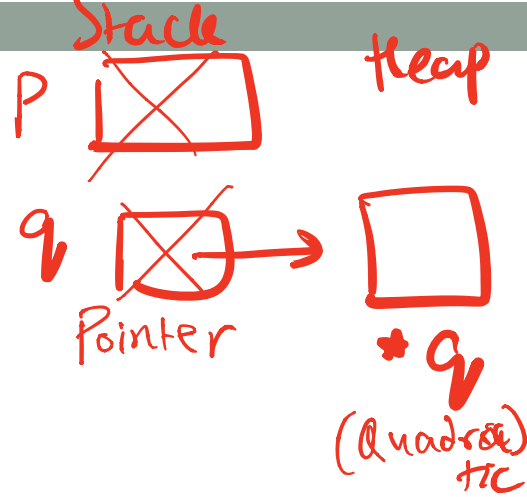
```
}
```

The default destructor is an empty routine.

We'll see the need to override the default in the next class (Linked Lists)

Destructor

```
void foo(){  
    Quadratic p;  
    Quadratic *q = new Quadratic;  
}
```



The destructor of which of the objects is called after foo() returns?

- A. p
- B. q
- C. *q
- D. None of the above

*q needs to be deleted explicitly

delete q; This statement will call the destructor then delete the object from heap.

Copy constructor

- Creates a new object and initializes it using an existing object

Copy constructor

- In which of the following cases is the copy constructor called?

A. Quadratic p1; Quadratic p2(1, 2, 3);

B. Quadratic p1(1, 2, 3); Quadratic p2(p1);

C. Quadratic *p1 = new Quadratic(1, 2, 3);

Quadratic p2 = *p1;

D. B&C

E. A, B & C

→ default c'tor

→ parameterized c'tor

→ copy c'tor

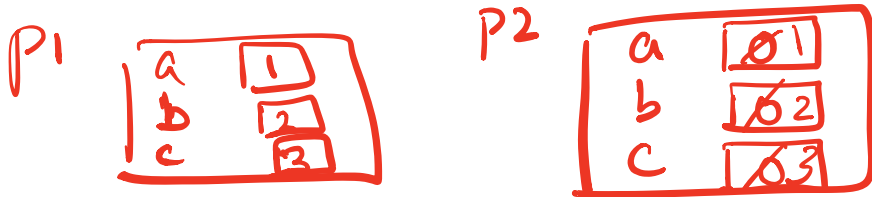
→ parameterized c'tor


← copy c'tor

Copy assignment

- Default behavior: Copies the member variables of one object into another

```
Quadratic p1(1, 2, 3); // Parametrized constructor  
Quadratic p2;  
p2 = p1; // Copy assignment function is called
```



```
double foo(Quadratic p){
    return p.evaluate(10);
}
int main(){
    Quadratic q(1, 2, 3);
    foo(q); 
}
```

Which of the following special methods is called as a result of calling foo?

- A. Parameterized constructor
- B. Copy constructor
- C. Copy Assignment
- D. Destructor

Summary

- ❑ Classes have member variables and member functions (method). An object is a variable where the data type is a class.
- ❑ You should know how to declare a new class type, how to implement its member functions, how to use the class type.
- ❑ Frequently, the member functions of an class type place information in the member variables, or use information that's already in the member variables.
- ❑ New functionality may be added using non-member functions, friend functions, and operator overloading *(next class)*

Next time

- Linked Lists + operator overloading