C++ CLASSES

```
class Integer {

 public:
   Integer();                        // Integer i;
   Integer(const Integer& i);        // Integer j(i);
   Integer(const std::string& s);    // Integer k(“123”);


   ~Integer(); // Destructor

   Integer& operator=(const Integer& i);  // assignment operator


}
```

C++ OBJECTS

```
Integer i; // Memory allocation on stack
Integer* i_ptr = new Integer(); // Memory allocation on heap
```

C++ OBJECTS

```
Integer i; // Memory allocation on stack

Integer* i_ptr = new Integer(); // Memory allocation on heap


// i_ptr is on heap – we need to take of memory deallocation
// Every call to new must be matched by a call to delete; failure to do
so causes memory leaks.
delete i_ptr;
```

DATA ENCAPSULATION IN C++

```
class Integer{
public:
    //! Default constructor
    /*! Setting up default integer. */
    Integer();

    /*!  Setter. */
    void setData(….);
    //..

private:
    // integer parameters
    DataType data;
}
```

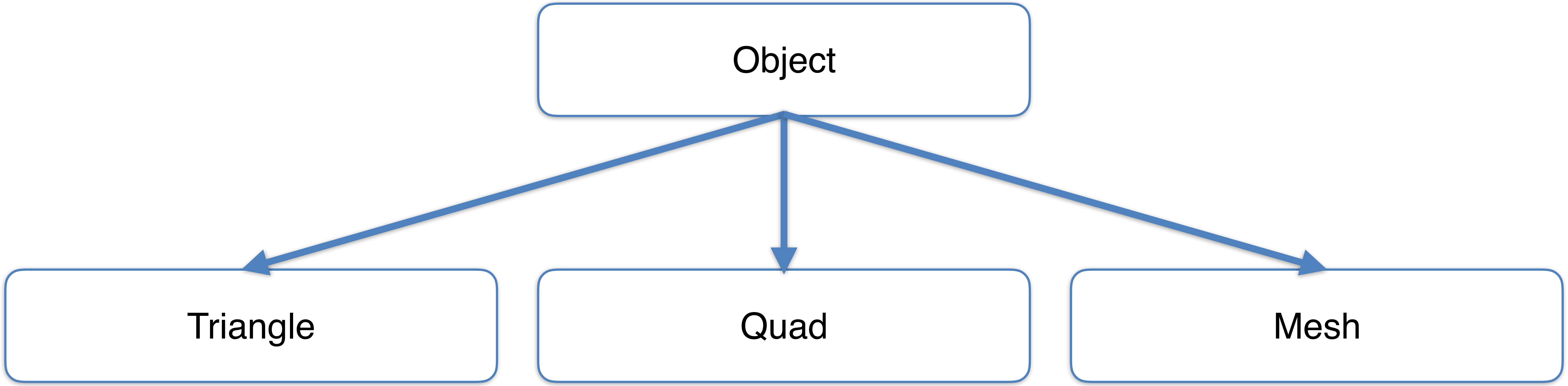
ACCESSING THE DATA MEMBERS

```
Camera i;
i.setData(…);
i.data = b;  // not allowed – private member
```

C++ OVERLOADING

```
// Arithmetic assignment operators
   Integer& operator+=(const Integer& i); // j += i;
   Integer& operator-=(const Integer& i); // j -= i;
   Integer& operator*=(const Integer& i); // j *= i;
   Integer& operator/=(const Integer& i); // j /= i;
   Integer& operator%=(const Integer& i); // j %= i;
```

C++ INHERITANCE



Object

Triangle

Quad

Mesh

```
class DerivedClass: access-specifier BaseClass
```

Quad Example:

```
class Quad: public Object
```