

Topics

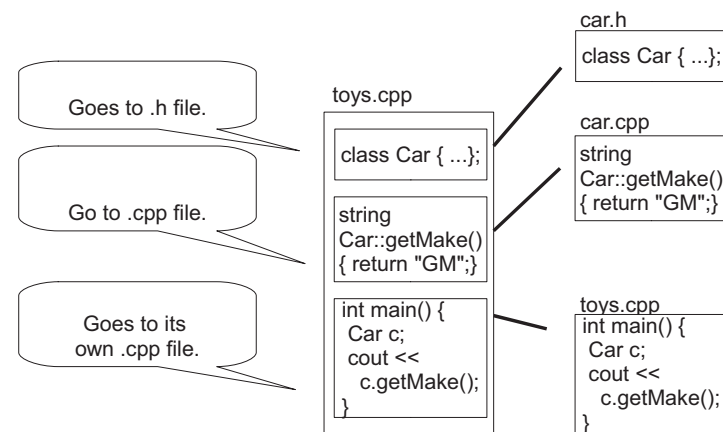
- 1) How can we organize our programs (into separate files)?
- 2) Can we use private member functions?

19/07/11

2

Breaking up a program

- We will break up programs into different parts:



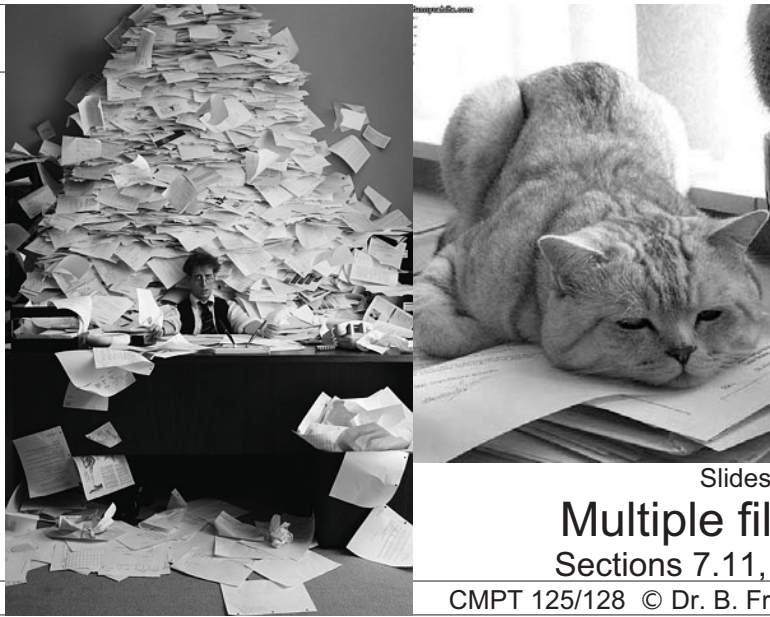
19/07/11

4

Programs in multiple files.

19/07/11

3



Multiple files

- Advantages of using multiple files:
 - Separate the... from the...
 - This supports encapsulation...
 - Helps keep code organized.
- Compiler (VC++) combines all files to create the executable program (.exe).
- - Other parts of the code access only the .h file.

19/07/11

carFull.cpp 5

Creating a class's .cpp file

- The class's .cpp file contains the...
 - It has to know the class definition (now in the .h file). So include the header file:
 - Note it's "...", not <...>
 - "." for files...
 - <...> for files... (so look in the compiler's directories for the file).

19/07/11

car.cpp 6

Creating a class's .h file

- Include guard:
 - Prevent a header file from...
- For file car.h, put at top:

```
#ifndef CAR_H_
#define CAR_H_
```
- Put at bottom:

```
#endif
```

.h Reads like:

If not defined CAR_H_
Then define CAR_H_
...
... Class declaration here
...
End of ifndef CAR_H_

Note the name CAR_H_ is arbitrary, you can use anything, just make it unique.

19/07/11

car.h 7

Using a class

- The Car class would be stored as:
 - car.h file: defines the class
 - car.cpp file: implements the class
- Create another file for the code that uses the class:
 - myProgram.cpp: Use the class as needed.
 - Must...
- This tells it enough about the Car class to use any part of it.

19/07/11

carSplit.cpp 8

```
#include <iostream>
#include <cmath>
using namespace std;
```

```
class Circle {
private:
    double radius;
public:
    void setRadius(double r){
        radius = r;
    }
    double getArea();
};
```

```
double Circle::getArea() {
    return 3.15 * pow(radius, 2);
}
```

```
int main() {
    Circle pizzaSmall;
    pizzaSmall.setRadius(6.0);
    cout << "Size of small: " << pizzaSmall.getArea() << endl;
    return 0;
}
```

Review

Show how this one file could be split into multiple files. Add any extra statements required!

19/07/11

9

Private member functions

19/07/11

10

Private member functions

- Member functions can be private:

```
#ifndef FUNNYNUM_H_
#define FUNNYNUM_H_

class FunnyNum {
private:
    long startNum;
    bool isOdd(long n);
public:
    FunnyNum(long start);
    long getNext();
};

#endif
```

```
#include "FunnyNum.h"
FunnyNum::FunnyNum(long start) {
    startNum = start;
}
long FunnyNum::getNext() {
    long next = startNum + 1;
    while (!isOdd(next)) {
        next ++;
    }
    return next;
}
bool FunnyNum::isOdd(long n) {
    return (n % 2 != 0);
}
```

19/07/11

11

Summary

- Split programs up into:
 - .h and .cpp for each class.
 - Ex: car.h, car.cpp, & apple.h, apple.cpp
 - .cpp for main() and non-member functions.
 - Ex: myCalculator.cpp
- Classes member functions can be:
 - public: Anyone can call them.
 - private: Only code within the class can call it.

19/07/11

12