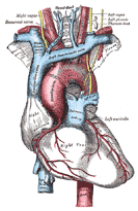
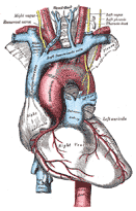
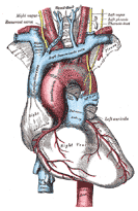


Anatomy of a Class

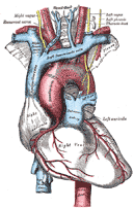


Part One: Specification (the .h file)

```
// flyingMonkey.h
// Outlines a Flying Monkey class
// Steve Martin, 6-25-01

#include <string>
using namespace std;

// spec for flyingMonkey class
class flyingMonkey {

        // Constructor
        flyingMonkey( );

        // Our methods for this function.
        // These should perform all tasks
        // involving the manipulation of
        // the data.
        bool isFlying( );

        // Data for this object
        string  monkeyName;
        bool monkeyAlive;
        bool monkeyFly;
};
```



Anatomy of a Class



Part Two: Implementation (the .cpp file)

```
// flyingMonkey.cpp
// Implements the Flying Monkey class described in flyingMonjey.h
// Steve Martin, 6-25-01

// include the spec
#include “flyingMonkey.h”

// constructor
flyingMonkey::flyingMonkey( ) {
        monkeyName = “Bob”;
        monkeyAlive = true;
        monkeyFly = false;
}

// Returns the current value of monkeyFly
bool flyingMonkey::isFlying( ) {
        return monkeyFly;
}
```

Questions to ponder (in case you don’t have any):

- *Where is “string.h” in the implementation file?*
- *What’s a constructor, and aren’t there different types?*
- *Why do I have a function JUST to return ‘monkeyFly’?*