1-answered before in chapter review.

2-

```
   #include <string>

#include <iostream>

class Person {
    private:
        static const int LIMIT = 25;
        std::string lname;
        char fname[LIMIT];
    public:
        Person() {lname = ""; fname[0] = '\0'; }
        Person(const std::string & ln, const char * fn = "Heyyou");
        ~Person();
        void Show() const;
        void FormalShow() const;
};
```

```
  #include "person.h"

  Person::Person(const std::string & ln, const char * fn)
  {
      lname = ln;
      strcpy(fname,fn);
  }

  Person::~Person()
  {
  }

  void Person::Show() const
  {
      std::cout << "n" << fname << " " << lname;
  }

  void Person::FormalShow() const
  {
      std::cout << "n" << lname << ", " << fname;
  }
```

```
#include "person.h"
int main()
{
    using namespace std;

    Person one;
    Person two("Smythecraft");
    Person three("Dimwiddy", "Sam");
    one.Show();
```

```
    cout << endl;
    one.FormalShow();
    two.FormalShow();
    two.Show();
    three.FormalShow();
    three.Show();

    cin.get();
    cin.get();
    return 0;
}
```

6-

```
class Move
{
   private:
      double x;
      double y;
   public:
      Move(double a = 0, double b = 0); // sets x, y to a, b
      void showmove() const; // shows current x, y values
      Move add(const Move & m) const;
      // this function adds x of m to x of invoking object to get new x,
      // adds y of m to y of invoking object to get new y, creates a new
      // move object initialized to new x, y values and returns it
      void reset(double a = 0, double b = 0); // resets x,y to a, b
};

#include "move.h"
#include <iostream>

Move::Move(double a, double b)
{
    x = a;
    y = b;
}

void Move::showmove() const
{
    std::cout << "x = " << x << ", y = " << y;
}

Move Move::add(const Move &m) const
{
    Move temp;
```

```
    temp.x = x + m.x;
    temp.y = y + m.y;

    return temp;
}

void Move::reset(double a, double b)
{
    x = a;
    y = b;
}
```