CS 246 F2003 Final

1. a) flock.h flock.cc animal.h dog.h sheep.h

b) Name mangling is when the compiler encodes function and variables names into unique names so that linkers can separate common names.

c) Not covered :P

2. a)

```
string monthname(int num) {
	unordered_map<int, string> months ({
		{1, "January"},
		{2, "February"},
		…
	});
	return months.at(num);
}
```

b)

```
int main() {
	int month, day, year;
	cin >> month >> day >> year;
	string monthString = monthname(month);
	cout << monthString << " " << day << " " << year << endl;
	return 0;
}
```

3. a) Cstrings are char arrays that end with a null byte, while strings are objects with automated memory management and control. Strings are generally safer, easier, and support different string manipulation functions.

b) No, no, yes

c) i. Does not compile

ii. Does not compile (did the old version of C++ just have * after string?)

iii. Does not compile

4. Not covered :P

5. a) i. iter = nums.begin();

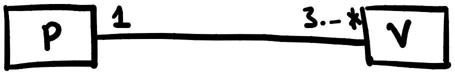
ii. ++iter;

iii. cout << *iter << endl;

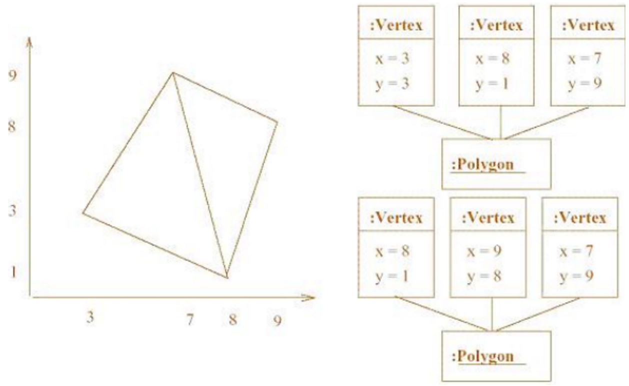
b)

```
int count = 0;
for (auto it = start; it != end; ++it) {
        if (*it < 0) {
                count += 1;
        }
}
return count;
```

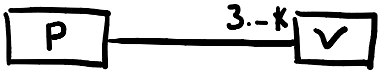
6. a)



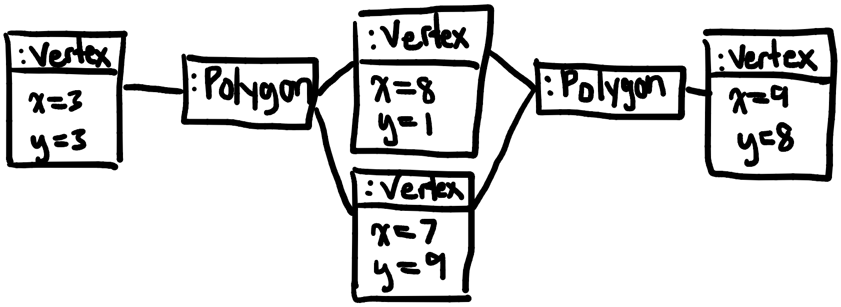
b)



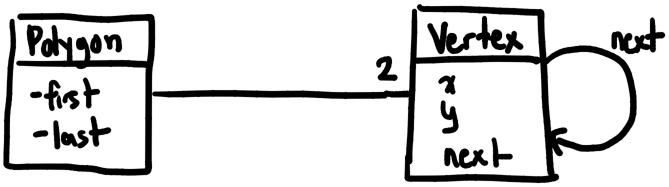
c)



d)



e)



7. Not covered :P

8. a)

```
Rect::Rect() {
    val[0] = 0;
    val[1] = 1;
}

int* Rect::getValue() {
    return val;
}

void Rect::putValue(int* v) {
    val = v;
}

void Rect::putValue(const Rect& v) {
    val[0] = v.val[0];
    val[1] = v.val[1];
}

Rect Rect::operator+(Rect& rhs) {
    Rect newRect;
    newRect.putValue(val + rhs.val);
    return newRect;
}
```

b)

```
void RSet::Add2Set(Rect b) {
    set.push_back(b);
}

void RSet::Sum(Rect& total) {
    for (const auto& rect : set) {
        total.putValue(total + rect);
    }
}
```

9. a) Fail: can't do na->&a

b) sorry: wrong number

c) -4

d) 0

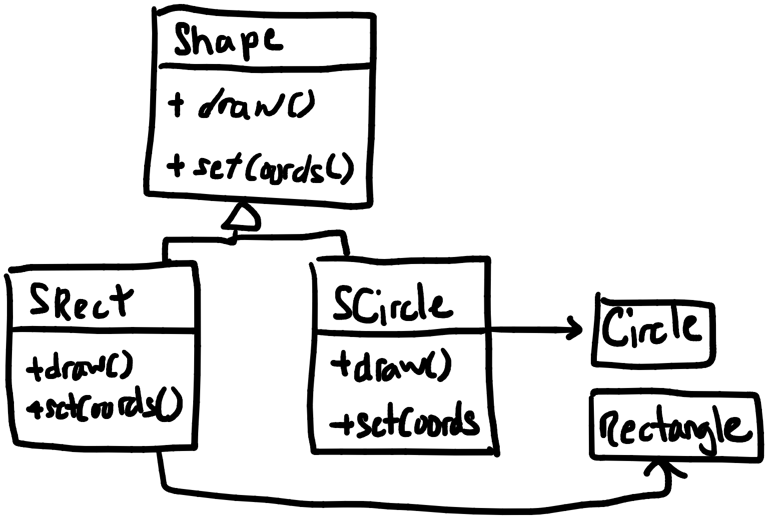
10. a)

```
template <class T>
class Set {
    vector<T> set;
public:
    void Add2Set(T t) {
        set.push_back(t);
    }
    void Sum(T &total) {
        for (auto t : set) {
            total.putValue(total + t);
        }
    }
};
```

b)

```
Set<Rect> ss;
...
```

11. a)



b)

```
s.setCoords(a, b);
s.draw();
```

c)

```
SRect::draw() {
    rectangle.display();
}

SRect::setCoords(int a, int b) {
    rectangle.setCorner(a, b);
}
```