

# CS 368 Announcements

## Wednesday, September 11, 2013

**Registered? Record your attendance on the sign-in sheet.**

**On waiting list? – Sign up on the yellow note pad.**

**TA lab hours will be posted tomorrow. Note they'll vary during the semester.**

### **Programming Assignment 1**

- has been assigned
- due Wednesday, September 25th

### **Last Time**

- historical overview
- simple C++ program: *simpleProg.cpp*
- edit-compile-run cycle in C++
  - edit: put code in text file ending in `.cpp`
  - compile (options):
    - A) `g++ simpleProg.cpp`
    - B) `g++ -Wall simpleProg.cpp -o simple`
  - run (corresponding options):
    - A) `a.out`
    - B) `simple`
- booleans
- constants
- enumerations

### **Today**

- structures
- arrays
- vectors
- *cardExample.cpp*

### **Next Time**

- start Ch. 3
- variables, references, pointers
- parameter passing
- pointer basics
- pointers to structs/classes

# Structs

## General form

```
struct struct-name {  
    data-declarations  
};
```

## Examples

```
struct Card {  
    Suits suit;  
    int rank;  
};
```

## Example

```
#include <iostream>
#include <vector>

using namespace std;

enum Suits      { clubs, diamonds, hearts, spades };
enum FaceCardRanks { jack = 11, queen = 12, king = 13,
                  ace = 1 };

struct Card {
    Suits suit;
    int rank;
};

void printCard(Card c) {
    switch (c.rank) {
        case 1:  cout << 'A'; break;
        case 11: cout << 'J'; break;
        case 12: cout << 'Q'; break;
        case 13: cout << 'K'; break;
        default: cout << c.rank; break;
    }

    switch (c.suit) {
        case hearts:  cout << 'H'; break;
        case spades:  cout << 'S'; break;
        case clubs:   cout << 'C'; break;
        case diamonds: cout << 'D'; break;
    }
}
```

## Example (cont.)

```
int main() {  
    Card c1, c2, c3;  
  
    c1.suit = hearts;  
    c1.rank = 5;  
  
    c2.suit = diamonds;  
    c2.rank = jack;  
  
    cout << "c1.suit is: " << c1.suit  
        << "\nc1.rank is: " << c1.rank << endl;  
  
    cout << "c1 is: ";  
    printCard(c1);  
    cout << endl;  
  
    cout << "c2.suit is: " << c2.suit  
        << "\nc2.rank is: " << c2.rank << endl;  
  
    cout << "c2 is: ";  
    printCard(c2);  
    cout << endl;  
  
    cout << "c3.suit is: " << c3.suit  
        << "\nc3.rank is: " << c3.rank << endl;  
  
    cout << "c3 is: ";  
    printCard(c3);  
    cout << endl;  
}
```

## **Arrays**

**Declaration (and creation)**

**Accessed as in Java**

**Warnings:**

## Example (cont.)

```
Card deck[4][13];

// Fill it
for (int s = 0; s < 4; s++) {
    for (int val = 1; val <= 13; val++) {
        deck[s][val-1].suit = Suits(s);
        deck[s][val-1].rank = val;
    }
}

// Print it out
cout << endl << "Printing 2-d array deck: " << endl;
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 13; j++) {
        printCard(deck[i][j]);
        cout << ", ";
    }
    cout << endl;
}
```

## Vectors

In standard library – need to `#include <vector>`

See [STL Reference](#) link under *Quick Links* for more info about vectors

**Declaration (and creation):**

```
vector<type> identifier;
```

**Example:**

```
vector<Card> deck2;

// Fill it
for (int s = 0; s < 4; s++) {
    for (int val = 1; val <= 13; val++) {
        deck2.push_back(deck[s][val-1]);
    }
}

// Print it out
cout << endl << "Printing vector deck: " << endl;
for (int i = 0; i < 52; i++) {
    printCard(deck2[i]);
    cout << ", ";
}
cout << endl;

return 0;
}
```