rand() &\& srand()

- rand()
  - used for random numbers from 0 to \(\text{RAND\_MAX}\)
  - \(\text{RAND\_MAX}\) defined constant in `<cstdlib>`

- srand()
  - used to “seed” the random number generator
  - seed off time
    - `#include <ctime>`
Random numbers between 0 - RAND_MAX

cout << "\n\nHow many random numbers do you wish to generate? ";
cin >> count;

while ( count > 0 ) {
    cout << rand() << endl;
    count = count - 1;
}

cout << "\nFinished!" << endl;
Random numbers between 0 - 1

cout << "\n\nHow many random numbers do you wish to generate? ";
cin >> count;

while ( count > 0 ) {
    cout << rand() * 1.0 / RAND_MAX << endl;
    count = count - 1;
}

cout << "\nFinished!" << endl;
Random numbers between 0 - 9

cout << "\n\nHow many random numbers do you wish to generate? ";
cin >> count;

while ( count > 0 ) {
    cout << rand() % 10 << endl;
    count = count - 1;
}

cout << "\nFinished!" << endl;
Seeding random numbers

• Run code again – get the same series of “random” numbers
#include <iostream>
#include <ctime>
#include <cctype>

using namespace std;

int main() {
    char response;
    cout << "Do you want a random seed? ";
    cin >> response;
    if ( toupper(response) == 'Y' ) {
        time_t seed = time(0);
        srand(seed);
    }
    cout << "\n\nHow many random numbers do you want? ";
    int count;
    cin >> count;
    while ( count > 0 ) {
        cout << rand() << endl;
        count = count - 1;
    }
    cout << "\nFinished!" << endl;
    return 0;
}
Problem: Lottery Game

* This program simulates one player playing a simple lottery game. The rules are:
  1) the player bets $1 each play, then chooses a number between 1 and MAX
  2) the computer draws a pseudo random number in same range
  3) if there is a match, the player wins $MAX
  4) the player keeps playing the game until either he/she is broke or wins once
int main ()
{
    int numPicked,        // number player picks
        numDrawn,        // number computer draws
        balance = START_BALANCE;  // start off with this amount on hand
    bool won = false;       // reset to true if player does win
    time_t seed;            // seeds (initializes) rand()

    seed = time (0);
    srand(static_cast<unsigned int>(seed));

    printIntro();
    printBalance(balance,0);

    while ( (balance != 0) && (!won) )
    {
        placeBet(balance);
        numPicked = pickNum(MAX);    // get human pick
        numDrawn = drawNum (MAX);     // computer draw
        checkForMatch(numPicked, numDrawn, won);
    }
    printClosing(won, balance);
    return 0;
}
void placeBet(int& bal)
{
    bal = bal - BET;
    printBalance(bal, 1);
}
int drawNum (int max) {
    int num;
    num = 1 + rand() % max;
    cout << "Number Drawn: " << num << endl;
    return (num);
}

int pickNum(int max) {
    int num;
    string junk;

    cout << endl << "Pick a number between 1 and " << max << " -> ";
    cin >> num;

    while ( (num < 1) || (num > max) || cin.fail() ) {
        if (cin.fail()) {
            cin.clear();
            getline(cin, junk);
        }
        cout << "Invalid number! Please try again -> ";
        cin >> num;
    }
    return num;
}
void checkForMatch(int pick, int drawn, bool& won) {
    if (pick == drawn) {
        won = true;
        cout << "The numbers match! You win!" << endl;
    } else {
        cout << "Sorry, no match." << endl;
    }
}
void printBalance(int bal, int bet)
{
    if (bet)
        cout << "\n\n*** Betting *** " << endl;

    cout << "Your balance is: $" << bal;
}