

```

////////////////////////////////////
// By: Frank Campos for C++ Module 1
// March 6, 2011
////////////////////////////////////

#include "cslotmachine.h"

CSlotMachine::CSlotMachine()
{
    game_on=false; //turn game off on start
    LOST_GAME=false; //turn off on start
    mPlayMaxcoin=false; //play max coin set to false on start

    mPlayCoin= 1;//set initial coin value to 1 on start
    mMaxcoin=1;//set max coin value to 1 on game start
    mCredits=INITIAL_AMOUNT;//set credit amount on start
    mPayout=0;//set the payout amount to 0 on start

    mRand1=0;//set random number 1 to 0
    mRand2=0;//set random number 2 to 0
    mRand3=0;//set random number 3 to 0
}

CSlotMachine::~CSlotMachine()
{
    //nothing to delete
}

void CSlotMachine::Init_Game()
{
}

void CSlotMachine::Spin()
{
    mRand1 = GetRandom(1, 9);
    mRand2 = GetRandom(1, 9);
    mRand3 = GetRandom(1, 9);

    int mPrice1=10*GetPlayCoin(); //price 1 all three numbers are 7
    int mPrice2=5*GetPlayCoin(); //price 2 three numbers are the same
    int mPrice3=3*GetPlayCoin(); //price 3 two number are the same
    int mLose = 1*-GetPlayCoin(); //lose

    int mCurrentPrice;//the actual price value

    if(mRand1==7 && mRand2==7 && mRand3==7)
    mCurrentPrice = mPrice1;
    else if (mRand1==mRand2 && mRand1==mRand3)
    mCurrentPrice = mPrice2;
    else if (mRand1==mRand2 || mRand1==mRand3 || mRand2==mRand1 || mRand2==mRand3
    || mRand3==mRand1 && mRand3==mRand2)
    mCurrentPrice = mPrice3;
    else
    mCurrentPrice = mLose;

    SetPayout(mCurrentPrice);//update payout
    SetCredit(mCurrentPrice);//update the current credit
}

```

```

}
int CSlotMachine::GetRandom(int mLow, int mHigh)
{
    return mLow + rand() % ((mHigh - 1) - mLow);
}
//return random number 1
int CSlotMachine::GetRandom1()
{
    return mRand1;
}
//return random number 2
int CSlotMachine::GetRandom2()
{
    return mRand2;
}
//return random number 3
int CSlotMachine::GetRandom3()
{
    return mRand3;
}
void CSlotMachine::GameRunning(bool game)
{
    game_on = game; //is the game on
}
void CSlotMachine::LostGame(bool status)
{
    LOST_GAME = status;
}
int CSlotMachine::GetCoin()
{
    return mPlayCoin;
}
int CSlotMachine::GetPlayCoin()
{
    if(isMaxCoinActive())
        return GetMaxCoin();
    else
        return GetCoin();
}
void CSlotMachine::ActivateMaxCoin(bool active)
{
    mPlayMaxcoin = active;
}
int CSlotMachine::isMaxCoinActive()
{
    return mPlayMaxcoin;
}
void CSlotMachine::SetCredit(int mCredit)
{
    mCredits += mCredit;
    if(mCredits<=0) {
        LostGame(true); //lost game
    }
}
int CSlotMachine::GetCredit()
{

```

```
        return mCredits;
    }
void CSlotMachine::SetMaxCoin(int mMcoin)
{
    if(mMcoin<11)//limit max coin to 10
        mMaxcoin=mMcoin;//set new max coin
}
int CSlotMachine::GetMaxCoin()
{
    return mMaxcoin; //return current max coin
}
void CSlotMachine::SetPayout(int mPay)
{
    if(mPay < 0)
        mPayout = 0;
    else
        mPayout = mPay;//set new payout amount
}
int CSlotMachine::GetPayout()
{
    return mPayout; //return current payout amount
}
```