```
// MONEY.CC

#include<iostream>
#include<iomanip>
using namespace std;

class Money
{
        private:
                int amount;
        public:
                Money();
                Money(int dollars, int cents);
                void add(int dollars, int cents);
                void set(int dollars, int cents);
                void print();
};

Money::Money()
{
        amount=0;
}
void Money::set(int dollars, int cents)
{
        amount=dollars*100+cents;
}
Money::Money(int dollars, int cents)
{
        set(dollars, cents);
}
void Money::add(int dollars, int cents)
{
        set(dollars, cents+amount);
}
void Money::print()
{
        cout << "$" << setw(1) << amount/100 << "."
                 << setw(2) << amount%100 << endl;
}

main()
{
        Money bob;
        Money tom(100,10);
        // HERE1
        bob.set(0,60);
        tom.add(4,95);
        bob.print();
        bob.add(1,110);
        bob.print();
        tom.print();
}


// CARDS.CC
#include<iostream>
using namespace std;

const int DEBUG=1;

struct Card
{
        int kind;   // Values 1 to 13 (Ace, 2, 3,...,J,Q,K)
        int suit;   // 1=Spade, 2=Heart, 3=Diamond, 4=club
        void printCard();
};

class Deck
{
        private:
                Card element[52];
                int locInDeck;
                Deck();
        public:
                Deck(int option);
                Card dealCard();
};

class Hand
{
        private:
                int NumCards;
        public:
                Card element[13];
                Hand();
                void addCard(Card c);
                Card playCard();
};

void Card::printCard()
{
        switch (kind)
        {
                case 11:
                        cout << "Jack";
                        break;
                case 12:
                        cout << "Queen";
                        break;
                case 13:
                        cout << "King";
                        break;
                case 1:
                        cout << "Ace";
                        break;
                default:
                        cout << kind;
        }
        cout << " of ";
        switch (suit)
        {
                case 1:
                        cout << "Spades";
                        break;
                case 2:
                        cout << "Hearts";
                        break;
                case 3:
                        cout << "Diamonds";
                        break;
                case 4:
                        cout << "Clubs";
                        break;
        }
}
```

```
Hand::Hand()
{
        NumCards=0;
}

void Hand::addCard(Card c)
{
        if(NumCards>12)
        {
                cerr << "Hand has too many cards!\n";
                exit(1);
        }
        element[NumCards]=c;
        NumCards++;
        if(DEBUG>1)
        {
           cout <<"Adding: ";
           c.printCard();
           cout << endl;
        }

}

Card Hand::playCard()
{
        Card tmp;
        if(NumCards<=0)
        {
                cerr << "Hand has too few cards!\n";
                exit(1);
        }
        NumCards--;
        tmp=element[NumCards];
        if(DEBUG>0)
        {
           cout <<"Playing: ";
           tmp.printCard();
           cout << endl;
        }
        return(tmp);

}


Deck::Deck()
{
}

Deck::Deck(int option)
{
        int i, j, tmp;
        int r1, r2;
        Card tcard;

        for(i=1;i<=13;i++)
                for(j=1;j<=4;j++)
                {
                        tmp=i+(j-1)*13;
                        element[tmp].kind=i;
                        element[tmp].suit=j;
                }
        locInDeck=52;

        if(option==1) // Shuffle
        {
                for(i=0;i<500;i++)
                {
                        r1=rand()%52;
                        r2=rand()%52;
                        tcard=element[r1];
                        element[r1]=element[r2];
                        element[r2]=tcard;
                }
        }
}
Card Deck::dealCard()
{
        if(locInDeck<=0)
        {
                cerr << "Dealing from empty hand!\n";
                exit(1);
        }
        locInDeck--;
        if(DEBUG>1)
        {
           cout <<"Dealing: ";
           element[locInDeck].printCard();
           cout << endl;
        }
        return(element[locInDeck]);

}

main()
{
        Hand N, S, E, W;
        Deck dealer(0);
        int i;
        Card nextC;

        for(i=0;i<4;i++)
        {
                nextC=dealer.dealCard();
                N.addCard(nextC);
                nextC=dealer.dealCard();
                W.addCard(nextC);
                nextC=dealer.dealCard();
                S.addCard(nextC);
                nextC=dealer.dealCard();
                E.addCard(nextC);
        }
        cout << "North plays: ";
        nextC=N.playCard();
        nextC.printCard();
        cout << endl << "West plays: ";
        nextC=W.playCard();
        nextC.printCard();
        cout << endl << "South plays: ";
        nextC=S.playCard();
        nextC.printCard();
        cout << endl;
}
```