```cpp
//
//  main.cpp
//  AbsoluteCpp_ch6_1
//

//Program to demonstrate the CDAccountV1 structure type.
#include <iostream>
using namespace std;

//Structure for a bank certificate of deposit:
struct CDAccountV1
{
    double balance;
    double interestRate;
    int term;//months until maturity
};

void getData(CDAccountV1 &theAccount);
//Postcondition: theAccount.balance, theAccount.interestRate, and
//theAccount.term have been given values that the user entered at the
 keyboard.

int main( )
{
    CDAccountV1 account;
    getData(account);

    double rateFraction, interest;
    rateFraction = account.interestRate/100.0;
    interest = account.balance*(rateFraction*(account.term/12.0));
    account.balance = account.balance + interest;

    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);
    cout << "When your CD matures in "
         << account.term << " months,\n"
         << "it will have a balance of $"
         << account.balance << endl;

    return 0;
}

//Uses iostream:
void getData(CDAccountV1 &theAccount)
{
    cout << "Enter account balance: $";
    cin >> theAccount.balance;
    cout << "Enter account interest rate: ";
    cin >> theAccount.interestRate;
    cout << "Enter the number of months until maturity: ";
    cin >> theAccount.term;
}
```

```cpp
//
//  main.cpp
//  AbsoluteCpp_ch6_2
//

//Program to demonstrate the CDAccount structure type.
#include <iostream>
using namespace std;

struct Date
{
    int month;
    int day;
    int year;
};

//Improved structure for a bank certificate of deposit:
struct CDAccount
{
    double initialBalance;
    double interestRate;
    int term;//months until maturity
    Date maturity; //date when CD matures
    double balanceAtMaturity;
};

void getCDData(CDAccount& theAccount);
//Postcondition: theAccount.initialBalance, theAccount.interestRate,
//theAccount.term, and theAccount.maturity have been given values.
//that the user entered at the keyboard.

void getDate(Date& theDate);
//Postcondition: theDate.month, theDate.day, and theDate.year
//have been given values that the user entered at the keyboard.

int main( )
{
    CDAccount account;
    cout << "Enter account data on the day account was opened:\n";
    getCDData(account);

    double rateFraction, interest;
    rateFraction = account.interestRate/100.0;
    interest = account.initialBalance*(rateFraction*(account.term/12.0));
    account.balanceAtMaturity = account.initialBalance + interest;

    cout.setf(ios::fixed);
    cout.setf(ios::showpoint);
    cout.precision(2);
    cout << "When the CD matured on "
         << account.maturity.month << "-" << account.maturity.day
         << "-" << account.maturity.year << endl
         << "it had a balance of $"
         << account.balanceAtMaturity << endl;
    return 0;
}

//uses iostream:
void getCDData(CDAccount& theAccount)
{
    cout << "Enter account initial balance: $";
    cin >> theAccount.initialBalance;
    cout << "Enter account interest rate: ";
    cin >> theAccount.interestRate;
    cout << "Enter the number of months until maturity: ";
    cin >> theAccount.term;
    cout << "Enter the maturity date:\n";
    getDate(theAccount.maturity);
}

//uses iostream:
void getDate(Date& theDate)
{
    cout << "Enter month: ";
    cin >> theDate.month;
    cout << "Enter day: ";
    cin >> theDate.day;
    cout << "Enter year: ";
    cin >> theDate.year;
}
```

```cpp
//
//  main.cpp
//  AbsoluteCpp_ch6_3
//

//Program to demonstrate a very simple example of a class.
//A better version of the class DayOfYear will be given in Display 6.4.
#include <iostream>
using namespace std;

class DayOfYear
{
public:
    int month;
    int day;
    void output( );
};

int main( )
{
    DayOfYear today, birthday;
    cout << "Enter today's date:\n";
    cout << "Enter month as a number: ";
    cin >> today.month;
    cout << "Enter the day of the month: ";
    cin >> today.day;
    cout << "Enter your birthday:\n";
    cout << "Enter month as a number: ";
    cin >> birthday.month;
    cout << "Enter the day of the month: ";
    cin >> birthday.day;

    cout << "Today's date is ";
    today.output( );
    cout << endl;
    cout << "Your birthday is ";
    birthday.output( );
    cout << endl;

    if (today.month == birthday.month && today.day == birthday.day)
        cout << "Happy Birthday!\n";
    else
        cout << "Happy Unbirthday!\n";
    return 0;
}


//Uses iostream:
void DayOfYear::output( )
{
    switch (month)
    {
        case 1:
            cout << "January "; break;
        case 2:
            cout << "February "; break;
        case 3:
            cout << "March "; break;
        case 4:
            cout << "April "; break;
        case 5:
            cout << "May "; break;
        case 6:
            cout << "June "; break;
        case 7:
            cout << "July "; break;
        case 8:
            cout << "August "; break;
        case 9:
            cout << "September "; break;
        case 10:
            cout << "October "; break;
        case 11:
            cout << "November "; break;
        case 12:
            cout << "December "; break;
        default:
            cout << "Error in DayOfYear::output. Contact software vendor.";
    }

    cout << day;
}
```

```
//
//  main.cpp
//  AbsoluteCpp_ch6_4
//

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

class DayOfYear
{
public:
    DayOfYear();
    DayOfYear(int newMonth, int newDay);
    void input( );
    void output( );
    void set(int newMonth, int newDay);
    //Precondition: newMonth and newDay form a possible date.

    void set(int newMonth);
    //Precondition: 1 <= newMonth <= 12
    //Postcondition: The date is set to the first day of the given month.

    int getMonthNumber( ); //Returns 1 for January, 2 for February, etc.
    int getDay( );

    bool operator < (DayOfYear b);

private:
    int month;
    int day;
};

int main( )
{
    DayOfYear today, bachBirthday;
    cout << "Enter today's date:\n";
    today.input( );
    cout << "Today's date is ";
    today.output( );
    cout << endl;

    bachBirthday.set(3, 21);
    cout << "J. S. Bach's birthday is ";
    bachBirthday.output( );
    cout << endl;

        if ( today.getMonthNumber( ) == bachBirthday.getMonthNumber( ) &&
               today.getDay( ) == bachBirthday.getDay( ) )
        cout << "Happy Birthday Johann Sebastian!\n";
    else
        cout << "Happy Unbirthday Johann Sebastian!\n";

    return 0;
}
```

```
//Uses iostream and cstdlib:
void DayOfYear::set(int newMonth, int newDay)
{
    if ((newMonth >= 1) && (newMonth <= 12))
        month = newMonth;
    else
    {
        cout << "Illegal month value! Program aborted.\n";
        exit(1);
    }
    if ((newDay >= 1) && (newDay <= 31))
        day = newDay;
    else
    {
        cout << "Illegal day value! Program aborted.\n";
        exit(1);
    }
}

//Uses iostream and cstdlib:
void DayOfYear::set(int newMonth)
{
    if ((newMonth >= 1) && (newMonth <= 12))
        month = newMonth;
    else
    {
        cout << "Illegal month value! Program aborted.\n";
        exit(1);
    }
    day = 1;
}


int DayOfYear::getMonthNumber( )
{
    return month;
}

int DayOfYear::getDay( )
{
    return day;
}

//Uses iostream and cstdlib:
void DayOfYear::input( )
{
    cout << "Enter the month as a number: ";
    cin >> month;
    cout << "Enter the day of the month: ";
    cin >> day;
    if ((month < 1) || (month > 12) || (day < 1) || (day > 31))
    {
        cout << "Illegal date! Program aborted.\n";
        exit(1);
    }
}

void DayOfYear::output( )
{
    switch (month)
    {
        case 1:
            cout << "January "; break;
        case 2:
            cout << "February "; break;
        case 3:
            cout << "March "; break;
        case 4:
            cout << "April "; break;
        case 5:
            cout << "May "; break;
        case 6:
            cout << "June "; break;
        case 7:
            cout << "July "; break;
        case 8:
            cout << "August "; break;
        case 9:
            cout << "September "; break;
        case 10:
            cout << "October "; break;
        case 11:
            cout << "November "; break;
        case 12:
            cout << "December "; break;
        default:
            cout << "Error in DayOfYear::output. Contact software vendor.";
    }

    cout << day;
}

bool DayOfYear::operator <(DayOfYear b){
    if ((month > b.month)|| ((month == b.month) && (day > b.day))){
        return false;
    } else {
        return true;
    }
}

DayOfYear::DayOfYear():month(1), day(1){

}

DayOfYear::DayOfYear(int newMonth, int newDay):month(newMonth), day(newDay){

    }
```

```
//
//  main.cpp
//  AbsoluteCpp_ch6_6
//

//===========================================================================
 ==
// Name        : Ch06-P12.cpp
// Author      :
// Version     :
// Copyright   : Your copyright notice
// Description :
/*
 * 12. Your Community Supported Agriculture (CSA) farm delivers a box of
  fresh fruits
and vegetables to your house once a week. For this Programming Project,
 define
the class BoxOfProduce that contains exactly three bundles of fruits or
 vegetables.
You can represent the fruits or vegetables as an array of type string. Add
 accessor
and mutator functions to get or set the fruits or vegetables stored in the
 array. Also
write an output function that displays the complete contents of the box on
 the
console.
Next, write a main function that creates a BoxOfProduce with three items
randomly selected from this list:
Broccoli
Tomato
Kiwi
Kale
Tomatillo
This list should be stored in a text file that is read in by your program.
 For now
you can assume that the list contains exactly five types of fruits or
 vegetables.
 */
//===========================================================================
 ==

#include <iostream>
#include <cstdlib>
#include <string>
using namespace std;

class BoxOfProduce{
public:
    void addProduce1(string produceName);
    void addProduce2(string produceName);
    void addProduce3(string produceName);
    string getProduce1();
    string getProduce2();
    string getProduce3();
    void displayBoxContent();
private:
    string produce[3];
};


int main() {

    BoxOfProduce box1;
    box1.addProduce2("Apple");
    box1.addProduce1("Celery");
    box1.addProduce3("Banana");
    box1.displayBoxContent();
    cout << "Item 2 in box is " << box1.getProduce2() << endl;


    return 0;
}

void BoxOfProduce::addProduce1(string produceName){

    produce[0] = produceName;
}

void BoxOfProduce::addProduce2(string produceName){

    produce[1] = produceName;
}

void BoxOfProduce::addProduce3(string produceName){

    produce[2] = produceName;
}


string BoxOfProduce::getProduce1(){
    return produce[0];
}

string BoxOfProduce::getProduce2(){
    return produce[1];
}

string BoxOfProduce::getProduce3(){
    return produce[2];
}

void BoxOfProduce::displayBoxContent(){
    for(string name:produce){
        cout << name << endl;
    }
}
```