

\*\*\* person.h \*\*\*

```
#ifndef PERSON_H
#define PERSON_H

#include <string>
using namespace std;
#include "date.h"

class Person
{
public:
    // constructors
    Person();
    Person( const string &name, const int id );
    Person( const string &name, const int id,
            const int year, const int month, const int day );

    // destructor
    ~Person();

    // accessor functions
    string get_name() const;
    void set_name( const string &name );
    int get_id() const;
    void set_id( const int id );

    void print_dateofbirth() const;
    // Note: could also overload '<<' to perform 'print'
    void print() const;

private:
    int id;
    string name;
    Date dateofbirth;// example of "composition" ie. declaring an
                    // object within another class
};

#endif
```

\*\*\* person.cpp \*\*\*

```
#include "person.h"
#include <string>
#include <iostream>
using namespace std;

// Note: for demonstration purposes only, this code has cout statements in the
// constructors and destructors; note how these are called when the code is
// executed. Such output is not normally placed inside of
// constructors/destructors.

Person::Person()
{
    name = "nobody";
    id = 0;
    cout << "executing Person::Person()" << endl;
}

Person::Person( const string &newname, const int newid ): name( newname ), id(
newid )
{
    cout << "executing Person::Person(const string &newname, “;
    cout << “ const int newid)” << endl;
}

Person::Person( const string &newname, const int newid,
                const int day, const int month, const int year ) :
                dateofbirth( day, month, year ), name( newname ), id( newid )
{
    cout << "executing Person::Person(const string &newname,“;
    cout << “const int newidconst,“;
    cout << "const int day,const int month,const int year) : “
    cout << “dateofbirth(day,month,year)), “
    cout << “ name( newname ), id( newid ) “ << endl;
}

Person::~Person()
{
    cout << "executing Person::~Person()" << endl;
}

string Person::get_name() const
{
    return name;
}
```

```

}

void Person::set_name( const string &newname )
{
    name = newname;
}

int Person::get_id() const
{
    return id;
}

Date Person::get_dateofbirth() const
{
    return dateofbirth;
}

void Person::print_dateofbirth() const
{
    dateofbirth.print(); // able to call Date public member function
}

void Person::print() const
{
    cout << "Name: " << name;
    cout << "\tId: " << id << "\tDate of birth: ";
    dateofbirth.print();
}

```

```

*** student.h ***
#ifndef STUDENT_H
#define STUDENT_H

#include "person.h"

class Student : public Person
{
public:
    // constructors
    Student();
    Student( const int gradyear );
    Student( const string &name, const int id, const int gradyear );
    Student( const string &newname, const int newid,
            const int newgradyear, const int day,
            const int month, const int year );

    // destructors
    ~Student();

    // accessors
    int get_gradyear() const;
    void set_gradyear( const int gradyear );

    void change_name( const string &newname );

    void print() const;

private:
    int gradyear; // expected graduation year of the student
};

#endif

```

**\*\*\* student.cpp \*\*\***

```
#include "student.h"
#include <iostream>
#include <string>
using namespace std;

Student::Student() : gradyear( 0 )
{
    cout << "executing Student::Student() : gradyear( 0 )" << endl;
}

Student::Student( const int newgradyear ) : gradyear( newgradyear )
{
    cout << "executing Student::Student( const int newgradyear ) :";
    cout << " gradyear( 0 )" << endl;
}

Student::Student( const string &newname, const int newid, const int
newgradyear ) : Person( newname, newid ), gradyear( newgradyear )
{
    cout << "executing Student::Student(const string &newname, ";
    cout << "const int newgradyear) : "
    cout << "Person(newname,newid),gradyear(newgradyear)" << endl;
}

Student::Student( const string &newname, const int newid, const int
newgradyear, const int day, const int month, const int year ) :
    Person( newname, newid, day, month, year ), gradyear( newgradyear )
{
    cout << "executing Student::Student(const string &newname, ";
    cout << "const int newgradyear) : "
    cout << "Person(newname,newid),gradyear(newgradyear)" << endl;
}
}
```

```
void Student::change_name( const string &newname )
{
    // name = newname; this does not compile since 'name'
    // is private in Person class

    set_name( newname ); // this command works since 'set_name' is
    // public in the person class
    // Note: 'change_name' is actually unnecessary since
    // Student objects will have access to the public
    // 'set_name' Person member fn
}

Student::~~Student()
{
    cout << "executing Student::~~Student()" << endl;
}

int Student::get_gradyear() const
{
    return gradyear;
}

void Student::set_gradyear( const int newgradyear )
{
    gradyear = newgradyear;
}

void Student::print() const
{
    Person::print();
    cout << "Grad year " << gradyear << endl;
    return;
}
}
```

**\*\*\* date.h \*\*\***

```
#ifndef DATE_H
#define DATE_H

// simple Date class
class Date
{
public:
    Date();
    Date( int init_day, int init_month, int init_year );
    ~Date();

    void print() const;

    int get_day() const;
    int get_month() const;
    int get_year() const;
    void set_day( const int );
    void set_month( const int );
    void set_year( const int );

private:
    int day;
    int month;
    int year;
};

#endif

*** date.cpp ***
#include <iostream>
using namespace std;
#include "date.h"

Date::Date() : day(1), month(1), year(1990)
{
    cout << "executing Date::Date()" << endl;
}

Date::Date( const int newday, const int newmonth, const int newyear ) :
    day(newday), month(newmonth), year(newyear)
{
    cout << "executing Date::Date( const int newday, const int newmonth,
const int newyear ) :";
```

```
        cout << " day(newday), month(newmonth), year(newyear)" << endl;
}

Date::~Date()
{
    cout << "executing Date::~Date()" << endl;
}

void Date::print() const
{
    cout << day << "-" << month << "-" << year << endl;
}

int Date::get_day() const
{
    return day;
}

int Date::get_month() const
{
    return month;
}

int Date::get_year() const
{
    return year;
}

void Date::set_day( const int newday )
{
    day = newday;
}

void Date::set_month( const int newmonth )
{
    month = newmonth;
}

void Date::set_year( const int newyear )
{
    year = newyear;
}
```

\*\*\* main.cpp \*\*\*

```
#include "person.h"
#include "student.h"
#include "undergrad.h"
#include <iostream>
using namespace std;
int main()
{
    Person person( "Bob", 99012345, 20, 10, 1980 );

    cout << "Person name: " << person.get_name() << endl;
    cout << "Person dateofbirth: ";
    person.print_dateofbirth();

    Student student( "Mary", 98111111, 9999, 20, 1, 1975 );

    cout << "Student name: " << student.get_name() << endl;
    cout << "Student id: " << student.get_id() << endl;
    cout << "Student dateofbirth: ";
    student.print_dateofbirth();

    person.set_name( "Bob II" );
    cout << "Person name: " << person.get_name() << endl;
    student.set_name( "Mary II" );
    cout << "Student1 name: " << student.get_name() << endl;
    student.change_name( "Mary III" );
    cout << "Student1 name: " << student.get_name() << endl;

    system("pause");
    return 0;
}
```

\*\*\* output \*\*\*

```
executing Date::Date( const int newday, const int newmonth, const int newyear ) :
day(newday), month(newmonth), year(newyear)

executing Person::Person(const string &newname,const int newidconst,const int
day,const int month,const int year) : dateofbirth(day,month,year))

Person name: Bob

Person dateofbirth: 20-10-1980

executing Date::Date( const int newday, const int newmonth, const int newyear ) :
day(newday), month(newmonth), year(newyear)

executing Person::Person(const string &newname,const int newidconst,const int
day,const int month,const int year) : dateofbirth(day,month,year))

executing Student::Student(const string &newname,const int newgradyear) :
Person(newname,newid),gradyear(newgradyear)

Student name: Mary

Student id: 98111111

Student dateofbirth: 20-1-1975

Person name: Bob II

Student1 name: Mary II

Student1 name: Mary III

executing Student::~~Student()

executing Person::~~Person()

executing Date::~~Date()

executing Person::~~Person()

executing Date::~~Date()

Press any key to continue
```