

## C++プログラミング III 実験／（後期）2022 年

出題日 2022年9月29日（木）

履修者は自分の担当者に CoursePower で提出. 提出〆切 10 月 5 日（水）**24:00**

課題 1 : CoursePower から prac01\_skel.cpp をダウンロードし, 適宜追加しプログラムを完成させなさい.

main 関数内で「今日の日付」と「名前と生年月日」を与えると, 実行例のように年齢と今年の誕生日までの日数などを出力するプログラムを作成したい. ここでは, うるう年かどうかは気にせずに**全てうるう年でない**としてよい. ただし, main 関数は変更しないこと.

```
$. /a.exe
今日: 2022/9/29.
Taro さんの誕生日 : 2002/9/29.
Peach-kun さんの誕生日 : 1923/3/3.
Jakob さんの誕生日 : 1654/12/27.

Taro さんは今 20 歳です
今日が誕生日です. お誕生日おめでとう !
Peach-kun さんは今 99 歳です
今年の誕生日は過ぎてしまいました
Jakob さんは今 367 歳です
誕生日まであと 89 日です
Jakob is released.
Peach-kun is released.
Taro is released.
```

```
//prac01_skel.cpp
#include <iostream>
using std::cout, std::cin, std::string;

// クラス Date 定義部分
// Date クラスに不備がある場合は訂正すること
class Date
{
private:
    int year; // 年
    int month; // 月
    int day; // 日
public:
    //コンストラクタ: 引数の値をデータメンバに格納
    Date(const int y, const int m, const int d)
    {
        year = y;
        month = m;
        day = d;
    }
}
```

```

// データメンバ year の値を返す
int get_year()
{
    //追加する
}
// データメンバ month の値を返す
int get_month()
{
    //追加する
}
// データメンバ day の値を返す
int get_day()
{
    //追加する
}
// プロトタイプ宣言
int days_from_newyearsday();
};
// データメンバに格納されている日にちについて元日からの日数を計算する
int Date::days_from_newyearsday()
{
    int days_in_month[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int days = day;

    // 終了条件が (month-1) であることに注意. (例)2月5日の場合には 31+5 になる
    for (int i = 0; i < (month - 1); i++)
    {
        days += days_in_month[i];
    }
    return days - 1; // 元旦からの日数なので 1 を引く. (例)1月1日の場合は 0
}
//クラス Person 定義部分
class Person
{
private:
    string name;    // 名前
    Date birth_day; // 生年月日
public:
    // コンストラクタ:引数の値をデータメンバに格納
    Person(const string fn, const Date d)
    {
        name = fn;
        birth_day = d;
    }
}

```

```

// データメンバ name の値を返す
string get_name()
{
    //追加する
}
// 名前と生年月日を出力する. (例)「Taro さんの誕生日: 2002/9/29.」を出力
void print_birth() const
{
    //追加する
}
int tobirthday(Date today); // 関数のプロトタイプ宣言
int ages(Date today);      // 関数のプロトタイプ宣言

// Person クラスのデストラクタの設定. 「名前+is released.」を表示する
//デストラクタの設定を追加する
};
// 引数から誕生日までの日数を計算し返す
// (「今日の日付」と「誕生日」との差を比較するため負の値になる可能性があることに注意)
int Person::tobirthday(Date today)
{
    //追加する
}
// 引数から誕生日までの年数を計算し返す
int Person::ages(Date today)
{
    // 今日と誕生日の日付を比較して, 今年誕生日がきているかどうかで場合分けする.
    //追加する
}
/*
今日が誕生日の場合: 「今日が誕生日です. お誕生日おめでとう!」
誕生日後の場合: 「今年の誕生日は過ぎてしまいました」
誕生日がまだ来ていない場合: 「誕生日まであと??日です」
を出力する
*/
void print_day(int date)
{
    //追加する
}
// main 関数部分
int main()
{
    Date today(2022, 9, 29); // 今日の日付
    Person taro("Taro", Date(2002, 9, 29));
    Person peach("Peach-kun", Date(1923, 3, 3));
    Person jakob("Jakob", Date(1654, 12, 27));

```

```
    cout << "今日: " << today.get_year() << "/" << today.get_month() << "/" <<
today.get_day() << ". ¥n";
    taro.print_birth();
    peach.print_birth();
    jakob.print_birth();

    cout << "¥n";
    cout << taro.get_name() << "さんは今 " << taro.ages(today) << " 歳です¥n";
    print_day(taro.tobirthday(today));
    cout << peach.get_name() << "さんは今 " << peach.ages(today) << " 歳です¥n";
    print_day(peach.tobirthday(today));
    cout << jakob.get_name() << "さんは今 " << jakob.ages(today) << " 歳です¥n";
    print_day(jakob.tobirthday(today));

    return 0;
}
```