**C++ Part II (INFO1-CE9265) Spring 2015 – Quiz 1**

Clement Chan

DayOfYear.h

#ifndef DayOfYear\_H

#define DayOfYear\_H

#include <iostream>

class DayOfYear{

private:

int day;

int month;

int year;

int day\_of\_year;

int \*days\_in\_month;

public:

DayOfYear(); //Default Constructor

DayOfYear(int Day, int Month, int Year); //Parameterized Constructor

bool leapyearcheck();

~DayOfYear(); //Destructor

DayOfYear(const DayOfYear &rhs); //Copy Constructor

DayOfYear& operator = (const DayOfYear &rhs);

//Calculate day of year

void calcdoy();

//Getters

const int getday();

const int getmonth();

const int getyear();

const int getdoy();

};

#endif

DayOfYear\_M.cpp

#include "DayOfYear.h"

DayOfYear::DayOfYear() : day(0), month(0), year(0), day\_of\_year(0) {

//Initializing the array with all the days in the months

days\_in\_month = new int[12];

for(int i=0; i<12; i++){

days\_in\_month[i] = 0;

}

}

DayOfYear :: DayOfYear(int Day, int Month, int Year) : day(Day), month(Month), year(Year), day\_of\_year(0){

//Initializing the array with all the days in the months

days\_in\_month = new int[12];

for(int i=0; i<12; i++){

days\_in\_month[i] = 0;

}

days\_in\_month[0] = 31; //Jan

if(leapyearcheck() == 0){ //Feb

days\_in\_month[1] = 28;

}else{

days\_in\_month[1] = 29;

}

days\_in\_month[2] = 31; //March

days\_in\_month[3] = 30; //April

days\_in\_month[4] = 31; //May

days\_in\_month[5] = 30; //June

days\_in\_month[6] = 31; //July

days\_in\_month[7] = 31; //August

days\_in\_month[8] = 30; //September

days\_in\_month[9] = 31; //October

days\_in\_month[10] = 30; //November

days\_in\_month[11] = 31; //December

if(month > 12 || month < 0){

std::cout<< "Month cannot be more than 12 or less than 0..." << std::endl;

exit(1);

}

if(day > days\_in\_month[month-1]){

std::cout<< "The Day entered exceed maximum days in the month..."<<std::endl;

exit(1);

}

}

DayOfYear::DayOfYear(const DayOfYear &rhs){ // Copy Constructor

std::cout<<"This is Copy Constructor ... " << std::endl;

day\_of\_year = rhs.day\_of\_year;

day = rhs.day;

month = rhs.month;

year = rhs.year;

days\_in\_month = new int[12];

for(int i=0; i<12; i++){

days\_in\_month[i] = 0;

}

for(int i=0; i<12; i++){

days\_in\_month[i] = rhs.days\_in\_month[i];

}

}

DayOfYear& DayOfYear::operator = (const DayOfYear &rhs){ //Assignment operator

std::cout<<"Using assignment operator ... " << std::endl;

day = rhs.day;

month = rhs.month;

year = rhs.year;

delete[] days\_in\_month;

days\_in\_month = new int[12];

for(int i=0; i<12; i++){

days\_in\_month[i] = 0;

}

for(int i=0; i<12; i++){

days\_in\_month[i] = rhs.days\_in\_month[i];

}

return \*this;

}

bool DayOfYear::leapyearcheck(){

if(year%4 == 0){

return true;

}else{

return false;

}

}

void DayOfYear::calcdoy(){

for(int i=0; i<(month-1); i++){

day\_of\_year += days\_in\_month[i]; //adding number of days each month passed by

}

day\_of\_year += day; //adding the additional days

}

//Getters

const int DayOfYear::getday(){

return day;

}

const int DayOfYear::getmonth(){

return month;

}

const int DayOfYear::getyear(){

return year;

}

const int DayOfYear::getdoy(){

return day\_of\_year;

}

DayOfYear::~DayOfYear(){

std::cout<<"Calling DayOfYear Destructor.."<<std::endl;

delete[] days\_in\_month;

}

Main.cpp

#include "DayOfYear.h"

int main(){

DayOfYear DOY(15,12,2016), DOY1(15,12,2015), DOY2;

DOY.calcdoy(); //Non-Leap Year

DOY1.calcdoy(); //Leap Year

std::cout<<"The number of days for " <<

DOY.getday()<<"/"<<DOY.getmonth()<<"/"<<DOY.getyear()<< " is " << DOY.getdoy() <<

std::endl;

std::cout<<"The number of days for " <<

DOY1.getday()<<"/"<<DOY1.getmonth()<<"/"<<DOY1.getyear()<< " is " << DOY1.getdoy()

<< std::endl;

//Check on assignment operators

DOY2 = DOY;

DayOfYear DOY3(DOY1);

DOY2.calcdoy();

std::cout<<"The number of days for (assignment operator) " <<

DOY2.getday()<<"/"<<DOY2.getmonth()<<"/"<<DOY2.getyear()<< " is " << DOY2.getdoy()

<< std::endl;

std::cout<<"The number of days for (copy constructor) " <<

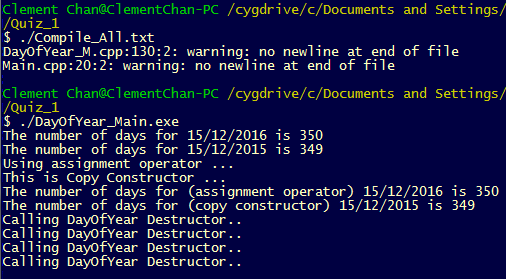
DOY3.getday()<<"/"<<DOY3.getmonth()<<"/"<<DOY3.getyear()<< " is " << DOY3.getdoy()

<< std::endl;

return 0;

}

**Output**



Compile File:

g++ -c DayOfYear\_M.cpp && g++ -c Main.cpp && g++ -o DayOfYear\_Main Main.o DayOfYear\_M.o