实验一 类与对象

Experiment One: Class and Object

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **项目** | **报告格式**  **Report format** | **代码质量**  **Code quality** | **注释质量**  **Comment quality** | **逻辑或思想描述**  **Necessitate logical description** | **独创性**  **Originality** | **合计**  **Total** |
| **百分比(%)**  **percentage** | **10** | **25** | **25** | **25** | **15** | **100** |
| 得分（score） |  |  |  |  |  |  |

**系 别(Department)： 计算机学院**

**专 业(Major)： 计算机类**

**班级姓名(Class & Name)：计类1710吴礼尉**

**学 号(Student ID)：2017011471**

**日 期(Data)： 2018.3.28**

**指导教师(Supervisor)：** 丁 濛

# 实验目的(Objects)

1. 了解编译器的使用方法，掌握利用命令行的方式进行编译和链接；Learn how to use C++ compiler to compile source code and link it to a binary program in a command-line style;
2. 学习引用型变量的使用方法，学会使用const关键字声明只读性变量，学习使用新关键字new和delete分配和删除内存空间；

Learn how to declare and use a variablewith type *reference*;

Learn how to use *const* prefix to declare constants with a specific type;

Learn how to use *new* and *delete* to allocate and release memory spaces from heap.

1. 学习和掌握C++对结构体的扩充；

Understand how C++ extended and modified the definition of the *struct* in C to provide the object-oriented feature.

1. 学习和掌握类的声明、定义方式以及如何声明、使用一个类的对象；验证构造函数、析构函数的调用时机；验证C++对于成员变量和函数的封装及访问控制；  
   Learn how to define a *class* and how to declare and use an object of a class;

Validate the mechanism of *constructor* and *destructor*: when they will be called automatically.

Validate how C++ encapsulates member properties and functions of a class and protects them from being accessed by outside the class anywhere.

1. 学习和掌握类的三种构造函数定义方式；拷贝构造函数的意义，默认拷贝构造函数的执行方式，何时需要显示地定义拷贝构造函数；

Learn how to define three different kinds of constructor function: Copy constructor, constructor without parameters (default constructor) and constructor w/ parameters;

Understand the essentials of a copy constructor, the way how a default copy constructor executes and when you need to define a copy constructor explicitly.

1. 学习和掌握类的对象的使用方法，包括自动分配和动态分配两种方式；

Learn how to declare and use an object of a class, in both dynamical and automatic way.

1. 学习和掌握std::cout 的基本使用方法；

Learn how to use *std::cout*.

1. 学习和体会面向对象思想中的抽象概念，即封装数据及对数据的操作；

Understand and practice the concept of abstract in OOP: Combine *data* and operations on *data* together as a new user-defined type.

# 实验内容 (Contents)

1. 配置编译环境，通过命令行方式对源代码进行编译和链接；

Configure the command-line compiling environment in your computer, which includes installing the compiler and configure the environment path. Try to compile a C++ source code and link it to a binary program using command-line.

1. 用指针的形式编写函数，void swap(int \*d1, int \*d2)，该函数实现交换两个整形变量的值；用引用的形式重写该函数，void swap(int &d1, int &d2)；可否使用const修饰上述函数的形参，即d1和d2？如果使用const修饰，会出现什么问题？

Write a function *void swap(int \*, int \*)* to swap the values of the given two parameters; Using *reference* type to rewrite that *swap* function. Can you apply *const* prefix to the two parameters? If you do that, what will happen?

1. 设计 Person 类。该类包括五个成员变量，分别是 姓名 char \*name，性别 char gender，出生年月日 int year, int month, int day。定义该类的三种构造函数，即 Person::Person(), Person::Person(const char \*, char, int, int, int), Person::Person(const Person& other)。定义该类的析构函数 Person::~Person(), 实现必要的内存清理功能。定义该类的公有成员函数 void Person::Print()，实现打印姓名，性别，出生年月日的功能。定义该类的公有成员函数 void Person::setName(const char \*)，实现对姓名的赋值；void Person::setGender(const char)，实现对性别的赋值；void Person::setBirthday(const int , const int , const int)，实现对出生年月日的赋值，需要考虑输入的有效性。  
    编写main函数，声明一个Person类的对象实例，并用自己的姓名、性别、出生年月日初始化该对象。验证析构函数、构造函数的调用时机。通过动态分配的方式实例化一个Person类的对象，并使用拷贝构造函数对该初始化该对象。

Design class Person.

The class will have 5 member variables, which are name (char \*), gender (char), birthday: year, month, day (int, int, int);

Define its three kinds of constructor: copy constructor, constructor without parameters (default constructor) and constructor w/ parameters.

Define its destructor function to do the necessary clean-up.

Define a public member function *void Person::Print()* to print the name, gender, birthday; a public member function *void Person::setName(const char \*)* to modify name, *void Person::setGender(const char)* to modify gender and *void Person::setBirthday(const int , const int , const int)* to modify birthday. You should consider the availability of the input.

Write a main function, define an object of class Person, and initialize it with your name, gender and birthday. Validate the time of the calling of constructor and destroyer. Define another object of Person in dynamical way and initialize it using copy constructor.

1. （附加题）编写函数，实现以字符串形式完成两个正整数的加法：输入为两个字符串，分别代表两个正整数，输出为一个字符串，表示最终结果，即char \*add\_string\_version(const char \*d1, const char \*d2)；编写main函数验证正确性。

例： char \*a = “0001200000320987630”, char \*b = “1”;

char \*result = long\_integer\_addition(a, b);

printf(“%s\n”, result); // 🡺 1200000320987631

(Bonus) Write a function to implement the addition of two positive integers expressed with type char\*, output a char\* expressing the sum. The prototype of that function can be *char \*long\_integer\_addition (const char \*d1, const char \*d2)*. Write a main function to validate your *long\_integer\_addition*.

Example:

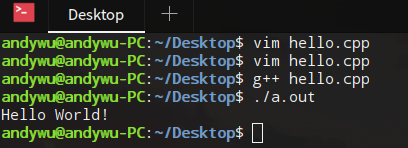
char \*a = “0001200000320987630”, char \*b = “1”;

char \*result = long\_integer\_addition(a, b);

printf(“%s\n”, result); // 🡺 1200000320987631

# 实验步骤 (Your steps or codes)

实验一



实验二

#include <cstdio>

void swap(int \*d1, int \*d2);

void swap(int &d1, int &d2);

int main(){

int a, b;

scanf("%d %d", &a, &b);

printf("a:%d, b:%d\n", a, b);

swap (&a, &b);

printf("a:%d, b:%d\n", a, b);

swap (a, b);

printf("a:%d, b:%d\n", a, b);

return 0;

}

void swap(int \*d1, int \*d2) //通过指针传递参数

{

int t;

t = \*d1;

\*d1 = \*d2;

\*d2 = t;

}

void swap(int &d1, int &d2) //

{

//通过引用传递参数时，不需要再加\*号，而是当做局部变量一样使用

//并且可以直接改变它们的值

int t = d1;

d1 = d2;

d2 = t;

}

实验三

#include <cstdio>

#include <iostream>

#include <cmath>

#include <cstring>

using namespace std;

//新建一个对象

class Person

{

private:

char \*name;

char gender;

int year;

int month;

int day;

public:

//默认构造函数

Person():year(1900), month(1), day(1)

{

}

//初始化构造函数

Person(const char \*name\_, const char gender\_, int year\_, int month\_, int day\_)

{

//用this指针来访问成员变量，并将其初始化

printf("Person(const char\*, char, int, int, int)\n");

this->name = new char [strlen(name\_)];

strcpy(this->name, name\_);

this->gender = gender\_;

this->year = year\_;

this->month = month\_;

this->day = day\_;

}

//拷贝构造函数

Person(const Person &other)

{

//用this指针来访问成员变量，并且通过.来访问参数传进来的引用

printf("Person(const Person&)\n");

this->name = new char [strlen(other.name)];

strcpy(this->name, other.name);

this->gender = other.gender;

this->year = other.year;

this->month = other.month;

this->day = other.day;

}

~Person()

{

printf("~Person()\n");

if (this->name != NULL)

{

delete [] this->name;

}

}

void Print()

{

printf("name:%s\n", this->name);

printf("gender:%c\n", this->gender);

printf("Birthday:%d-%02d-%02d\n", this->year, this->month, this->day);

printf("\n");

}

void setName(const char \*name)

{

if (this->name != NULL)

{

delete [] this->name;

}

this->name = new char [strlen(name)];

strcpy(this->name, name);

}

void setGender(const char gender)

{

this->gender = gender;

}

void setBirthday(const int year, const int month, const int day)

{

this->year = year;

this->month = month;

this->day = day;

}

};

int main(){

char name[100];

int year, month, day;

char gender;

scanf("%s", name);

scanf("%d %d %d", &year, &month, &day);

scanf(" %c", &gender);

Person p(name, gender, year ,month, day);

memset(name, 0, 100);

p.Print();

Person s (p);

s.Print();

s.setName("Mark");

s.setGender('F');

s.setBirthday(1998, 11, 29);

s.Print();

p.Print();

Person \*r = new Person(s);

r->Print();

delete r;

return 0;

}

附加题

/\*

这个程序，先将字符串读入，然后创建两个数组，用来逆序存放两个字符串每位数上对应的数值。

\*/

#include <cstdio>

#include <cstring>

#include <cstdlib>

int max(int a, int b);　//求最大值函数

void print(const int \*a, const int n);　//打印函数

void set(int \*a, char \*s, int len, int n);　//将字符串装进数组里

void add(int \*a1, int \*a2, int \*sum, int len);　//两个数组相加

int main()

{

//声明变量

char s1[100], s2[100];

int a1[100], a2[100], sum[101];

//输入字符串

fgets(s1, 100, stdin);

fgets(s2, 100, stdin);

//获取字符串长度

int len1 = strlen(s1) - 1;

int len2 = strlen(s2) - 1;

//取最长的那个

int lenmax = max(len1, len2);

//初始化数组

memset(a1, 0, sizeof(a1));

memset(a2, 0, sizeof(a2));

//调用set函数，将字符串对应填入数组里

set(a1, s1, lenmax, len1);

set(a2, s2, lenmax, len2);

//两个数组相加

add(a1, a2, sum, lenmax);

//打印

print(a1, lenmax);

print(a2, lenmax);

print(sum, lenmax + 1);

return 0;

}

int max(int a, int b)

{

//返回最大值

return a > b ? a : b;

}

void print(const int \*a, const int n)

{

//输出数组，并且前面的０不输出

int flag = 0;

for (int i = 0; i < n; i++)

{

if(a[i] != 0)

{

flag = 1;

}

if(flag)

printf("%d ", a[i]);

}

printf("\n");

}

//len表示最大字符串长度，n表示当前数组长度。

void set(int \*a, char \*s, int len, int n)

{

int i, j;

//从最末端依次读取

for (i = len - 1, j = n - 1; i >= 0 && j >= 0; i--, j--)

{

//每个字符＆１５就等于其对应的数字

a[i] = s[j] & 15;

}

}

//两个数组相加

void add(int \*a1, int \*a2, int \*sum, int len)

{

int t = 0;　// t表示进位

for (int i = len - 1, j = len; i >= 0; i--, j--)

{

int temp = (a1[i] + a2[i] + t);

//temp　表示当前两个数组对应位数和上一位的进位相加的值

sum[j] = temp % 10;//取一位，应该取模

if (temp >= 10)

{

t = １;//如果相加大于１０则进１

}

else

{

t = 0;　//否则进０

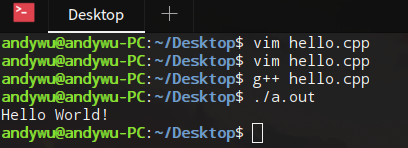
}

}

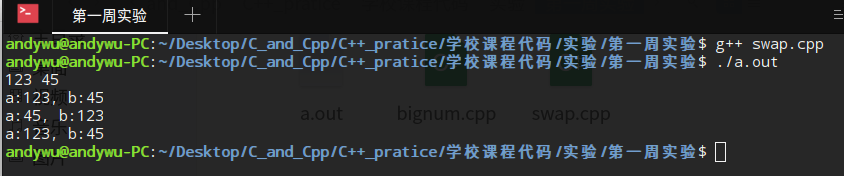
}

# 实验结果 (Experimental results)

实验一



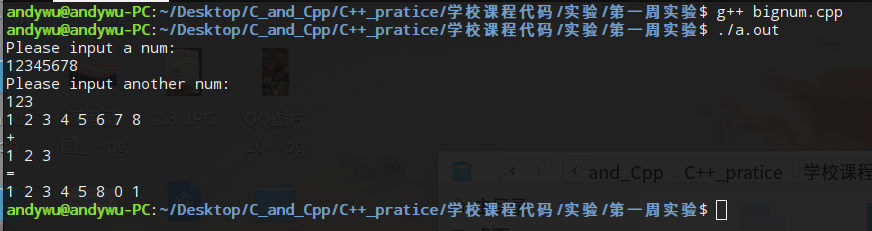
实验二



实验三



附加题



# 结论 (Conclusion)

1. 命令行可以进行c++文件的编译
2. 可以通过引用来访问并修改参数的值
3. 创建一个对象，编写对应的成员函数，可以对其成员变量进行操作。
4. 当需要操作的数字比较大的时候，可以考虑用字符串将其读入，再转换成为数组，然后进行相加运算，其中相加运算需要写一个算法来实现。