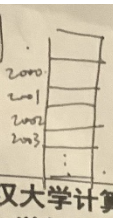
3. 没内存间如图



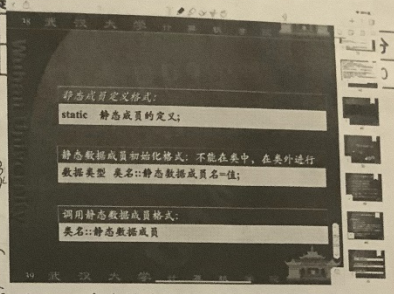
武汉大学计算机学院
2019-2020 学年度第 2 学期 2019 级
《高级语言程序设计》期末考试试卷 A

姓名：_____ 学号：_____ 班级：_____

说明：**开卷考试**，答案请全部写在答题纸上，写在试卷上无效。

未经主考教师同意，考试试卷、答题

题号	一	二	
总分	16	10	

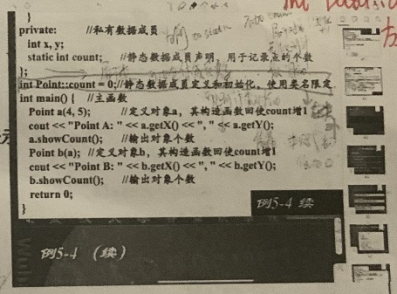


一. 简答题：（共 4 小题，每小题 4 分，

1. （4 分）设有下面的类定义：

```
class Data{
  public:  int num;
           static int data;
};
```

请简要说明对类 Data 来说，成员 num 和 data 二者有什么区别？请写出对成员 data 进行定义和初始化为 0 的语句，并说明该语句放置的正确位置。



2. （4 分）设有如下的定义：

```
unsigned int x1=0X67FE, x2=0x4E79, x3=1;
```

请指出表达式（x1 & x2 + x3）的结果，用十六进制表示

3. （4 分）假设有如下的定义：

```
char s[50]="The C++ Programming Language!";
short *p1=(short*)(s+4),*p2=(short *)(s+20);
```

请指出表达式 p2-p1 的值为多少，为什么？

4. （4 分）什么是类的封装性？C++中类的封装性是如何实现的？

二. 分析改错题：（共 2 小题，共 10 分）

5. （6 分）下列程序片段从 1 乘到 50，虽然通过了语法检查，但测试工程师认为程序存在 2 个错误，1 个稳定性隐患。请您找出它们并改正：

```
1.   long int s = 0;
2.   for (int i = 1; i!=51; i++)
3.     s *= i;
```

6. （4 分）下列程序片段中函数判断形参 num 是否为素数（0 为非素数）。请指出该函数可以优化的地方,并改正：

```
1.   int prime(int num)
2.   {  int val = 0;
3.     for (int n=2;  n <=(num/2);  n++)
```

```
4.          if (!(num % n)) val++;
5.          return (val!=0);
6.      }
```

三. 程序阅读与分析题：（共 2 小题，每小题各 12 分，共 24 分）

7. （12 分）请仔细阅读以下程序，完成下列三个任务：

1) 说明程序的用途和功能，以及调用层次

2) 已知数组 arr 在内存初始状态的示意为：

3	7	1	6	9	4	8	5

请照此样式，画出 dataprocess () 函数执行时 arr 的数据在内存中变化过程示意，并写出程序的最终输出结果。

3) 请改写函数 dataprocess () 为非递归形式。

```
1.    void dataprocess(int a[], int n)
2.    {
3.       int tmp;
4.       for (int i = 0; i < n; i++)
5.       {
6.          if (a[i] < a[n - 1])
7.          {
8.             tmp = a[i];
9.             a[i] = a[n - 1];
10.            a[n - 1] = tmp;
11.         }
12.
13.      }
14.      if (n > 2)
15.      {
16.         dataprocess(a, n - 1);
17.         n--;
18.      }
19.   }
20.
21.   int main()
22.   {
23.      int arr[8] = {3,7,1,6,9,4,8,5};
24.      int n = 8;
25.      dataprocess(arr, n);
26.      for (int i =0; i < n; i++)
27.      {
28.         cout << arr[i] << " ";
29.      }
30.      return 0;
31.   }
```

8. （12 分）请仔细阅读以下程序，完成下列两个任务：

1) 请简要描述程序中各类之间的关系，并阐述程序的执行过程。

2) 请给出程序的输出结果。

```
1.    class A {
2.    public:
3.       A(int i):va(i) {
```

```
        cout << "Constructing A " << i << endl;
        count++;
        cout << "Object A count = " << count << endl;
    }
private:
    int va;
    static int count;

};
int A::count = 0;

class B:public A {
public:
    B(int i,int j):A(i),vb(j){
        cout << "Constructing B " << i<<" " << j<< endl;
    }
private:
    int vb;

};

class C {
public:
    C(int i):vc(i) {
        cout << "Constructing C " << i<< endl;
    }
private:
    int vc;
};

class D: public C, public B {
public:
  D(int a, int b, int c, int d, int e, int f) : B(a,b), objb(d,e), objc(f),C(c)
  { }
private:
    C objc;
    B objb;
};

int main() {
    D obj(1, 2, 3, 4, 5, 6);
    return 0;
}
```

(1) AB之间 BC之间
B为派生类 D为派生类
A为基类 BC为基类
B对A方式：公有 对A方式：公有

程序执行先程：

先赋值. B(1,2) objb(4,5) objc(6) C(3).

C(3) 执行 (在C类中)车间出.

Constructing C 3

objc(6) ? 按顺序执行

C(3). → C C3.
B(1,2) ⇒ 先执行 A1→…1 →count
再执行.C B…2
objc(6) → C C6.
objb(4,5) → A. 4 count 2
ConB 45

(2)结果： Con. C3.
Con… A1
Object A count=1
Con… B12
Con… C6
Con… A4
Object A count=2.
Con… B45.

四. 编程实现题（共2小题，每小题各15分，共30分）

9. （15分）请编程完成函数 priceStatistics()的代码，该函数功能是给定一组商品价格数据，计算其中的最高、最低和平均价格。

例如，有如下的定义：

float price[10]={7.7,10,6.7,5.4,9.2,3.4,6.5,9.9,8.7,9}; //商品价格数组

float max, min, avg; //统计变量，max 对应最高价格，min 为最低价格，avg 为平均价格

则当函数调用 priceStatistics(price,10,max,min,avg) 执行后，变量 max、min、avg 取值分别为 10、3.4、7.65。

说明：

（1）请自行补充完善所需的主函数或辅助函数；

（2）在主函数中调用 priceStatistics (price,10,max,min,avg);

(3) 最高、最低价格和平均价格数据类型均为 float。

10. (15 分) 请完整定义和实现一个采用 24 小时制计时的时钟类 Clock，要求 Clock 类支持如下操作：

```
Clock c1;                  //c1 的 hour 为 0，minute 为 0，second 为 0
Clock c2(13,40,40);        // c2 的 hour 为 13，minute 为 40，second 为 40
c1+c2;                     //计算两个 Clock 对象 c1 和 c2 的和，返回 Clock 对象
c1<c2;                     //判断两个 Clock 对象 c1 和 c2 的大小（时间先后），返回 bool 值
                           //例如：8:20:21 < 13:40:40，返回值为 true
++c2;                      //c2 时间加 1 秒

class Clock{
private:
   int hour, minute, second;
public:
      //请补充 Clock 的定义
};
```

五. 程序分析与设计题（共 20 分）

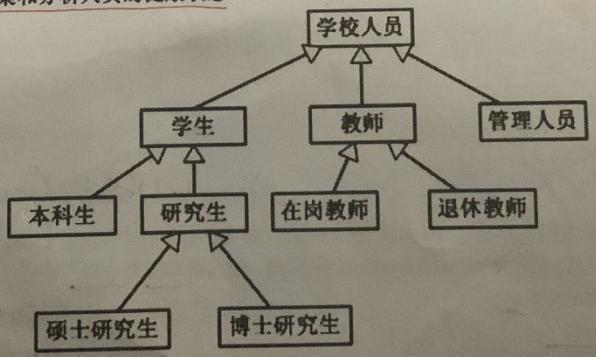
说明：

A. 本大题以说明分析设计思路为主，不必写出函数的完整实现代码，完整函数代码将不作为评分依据；

B. 对于其中的函数定义，要求注释说明其功能、形式参数的含义和类型、返回值类型等；

C. 设计中所需的重要数据类型的定义要有注释说明。

11. (20 分) 分析设计出健康信息登记管理模块：下图是某大学人员的分类图，现需要对各类人员的信息进行登记，每日收集和分析人员的健康状态。



此模块能够实现的业务功能包括：

1) 登记各类人员的基本信息,如：编号、姓名、性别、手机号码、证件类型（工作证、学生证、退休证）、证件号码等；

2) 登记居住信息，如：当前所在国家/地区、当前省份、当前城市、当前社区、详细住址等；

3) 登记每日健康信息，如：登记时间、体温、健康状态（健康、发热、疑似、确诊、治愈、其它）等；

4) 统计分析功能，包括：统计全体人员各种健康状态人数的日报和周报；统计每类人员各种健康状态人数的日报和周报；统计每周居住地发生变化的人数。

需要设计完成的任务有以下三项：

1) 设计出此管理模块所需的类，可用 UML 类图说明所设计的类间关系，并请简要说明你给出的类设计方案的理由和优缺点。

2) 请用规范的 C++类定义语法，写出上述所设计类的定义语句。根据所需情况写出类的数据成员、函数成员、构造函数和析构函数的声明，以及成员访问权限。所有函数的实现（函数体）语句不必写出来。

3) 设计出能完成上述信息登记、统计分析等业务功能所需的函数。你的设计是把这些函数定义为类的成员函数，还是非成员函数呢？请简要说明你的设计理由。
对所设计的这些函数请写出函数的声明语句，所有函数的实现（函数体）语句不必写出来，但请注释说明函数功能、形式参数的含义和类型、返回值类型等必要信息。

9. 主: int main(){
float price[10]={---,---};
float max, float min, float avg;
price(statics(price, 10, max, min, avg)
cout<<"max="<<max<<endl;
return 0;
}

Ans 2. 0x6FFE = 0110 0111 1111 1110

0x4E79 = 0100 1110 0111 1001

位与得 0100 0110 0111 1000

1: + 0000 0000 0000 0001

0100 0110 0111 1001

[illegible] 0x4679

3.

4. 封装就是把对象的属性和行为结合成一个独立的单位并尽可能隐藏对象的内部细节

其有两个含义：封装性：把对象全部属性行为结合在一起形成一个不可分割的独立单位

信息隐蔽：尽可能隐蔽对象的内部细节对外形成一道屏障，与外部联系只

能通过外部接口实现。

好处：对象使用者，设计者分开，提供代码复用性，减少开发软件系统难度。

9.

```
#include <iostream>
#include <cmath>
using namespace std;
void priceStatics(const float price[], int n, float &max, float &min, float &avg){
    max=0.0  //初始化
    for (int i=0; i<n; i++){
        if (price[i]>max){
        max= price[i];
        }
    }
    }
    min=100.0
    for (int i=0; i<n; i++){
        if (price[i]<min){
        min= price[i]
        }
    }
    float sum=0.0
    avg=0.0
    for (int i=0; i<n; i++){
        sum+ = price[i];
    }
    avg = sum/n
    return;
}
```

年函中给出计算公式，而并非输出，这变量
max. min. avg 在主函数里定义！ 三个变量 ，来输出
若辅助要用 地址引用 一切在主！
应在循环前

10題 正確答案:

```
#include<iostream>
using namespace std;
class Clock {
public:Clock(int hour = 0, int minute = 0, int second = 0);
  void showTime();
  Clock operator+(Clock& c2);
  Clock& operator++();
  void showTime(Clock c);
private:
 int hour, minute, second;
};
Clock::Clock(int hour, int minute, int second) {
 if (0 <= hour && hour < 24 && 0 <= minute && minute < 60 && 0 <= second && second < 60) {
  this->hour = hour;
  this->minute = minute;
  this->second = second;
 }
 else
  cout << "Time error" << endl;
}
Clock Clock::operator+( Clock& c2) {
 Clock c(hour + c2.hour, minute + c2.minute, second + c2.second);
 if (second >= 60) {
  second -= 60;
  minute++;
  if (minute >= 60) {
   minute -= 60;
   hour = (hour + 1 % 24);
  }
 }
 return c;
}
Clock& Clock::operator++() {
 second++;
 if (second >= 60) {
  second -= 60;
  minute++;
  if (minute >= 60) {
   minute -= 60;
   hour = (hour + 1 % 24);
  }
 }
 return *this;
}
void Clock:: showTime(Clock c){
      int a,b,d;
      a = c.hour;
      b = c.minute;
      d = c.second;
      cout<<a<<":"<<b<<":"<<d<<endl;
}

int main() {
 Clock c1;
 cout << "c1=";
 c1.showTime(c1);
 Clock c2(13, 40, 40);
 cout << "c2=";
 c2.showTime(c2);
```

bool operator < (const Clock & c_2) const {
return (hour < c_2.hour) || (hour = c_2.hour && minute < c_2.minute) || (hour = c_2.hour && minute = c_2.minute && second < c_2.second);
}

10题 正确答案：

```
#include<iostream>
using namespace std;
class Clock {
public:Clock(int hour = 0, int minute = 0, int second = 0);
   void showTime();
   Clock operator+(Clock& c2);
   Clock& operator++();
   void showTime(Clock c);
private:
 int hour, minute, second;
};
Clock::Clock(int hour, int minute, int second) {
 if (0 <= hour && hour < 24 && 0 <= minute && minute < 60 && 0 <= second && second < 60) {
  this->hour = hour;
  this->minute = minute;
  this->second = second;
 }
 else
  cout << "Time error" << endl;
}
Clock Clock::operator+( Clock& c2) {
 Clock c(hour + c2.hour, minute + c2.minute, second + c2.second);
 if (second >= 60) {
  second -= 60;
  minute++;
  if (minute >= 60) {
   minute -= 60;
   hour = (hour + 1 % 24);
  }
 }
 return c;
}
Clock& Clock::operator++() {
 second++;
 if (second >= 60) {
  second -= 60;
  minute++;
  if (minute >= 60) {
   minute -= 60;
   hour = (hour + 1 % 24);
   }
  }
 return *this;
}
void Clock:: showTime(Clock c){
        int a,b,d;
        a = c.hour;
        b = c.minute;
        d = c.second;
        cout<<a<<":"<<b<<":"<<d<<endl;
}

int main() {
 Clock c1;
 cout << "c1=";
 c1.showTime(c1);
 Clock c2(13, 40, 40);
 cout << "c2=";
 c2.showTime(c2);
```

```
 Clock c3;
 c3 = c1 + c2;
 cout << "c3=c1+c2=";
 c3.showTime(c3);
 cout << "++c2=";
 (++c2).showTime(c2);
 return 0;
}
```

未比较大小

```
class Member
protected:
    string number, name, phoneNumber, IDNumber;
    bool gender;
    int idType;
    string country, province, city, community, detailedAddr;
    bool isAddrChanged;
    int time, temperature;
    vector<int> state;
public:
    Member(){}
    void setInfo(string num, string name, string pNum, string IDNum, bool gender, int idType);
    void setAddr(string country, string province, string city, string community, string detailedAddr);
    void setTemperature(int time, int temperature, int stateNum, int i);
    friend void totalHealthState(Member p[], int size, int i);
    friend int addrChangedNumber(Member p[]);
};
class Student :public Member{};
void totalHealthState(Member p[], int size, int i);
int addrChangedNumber(Member p[]);
```

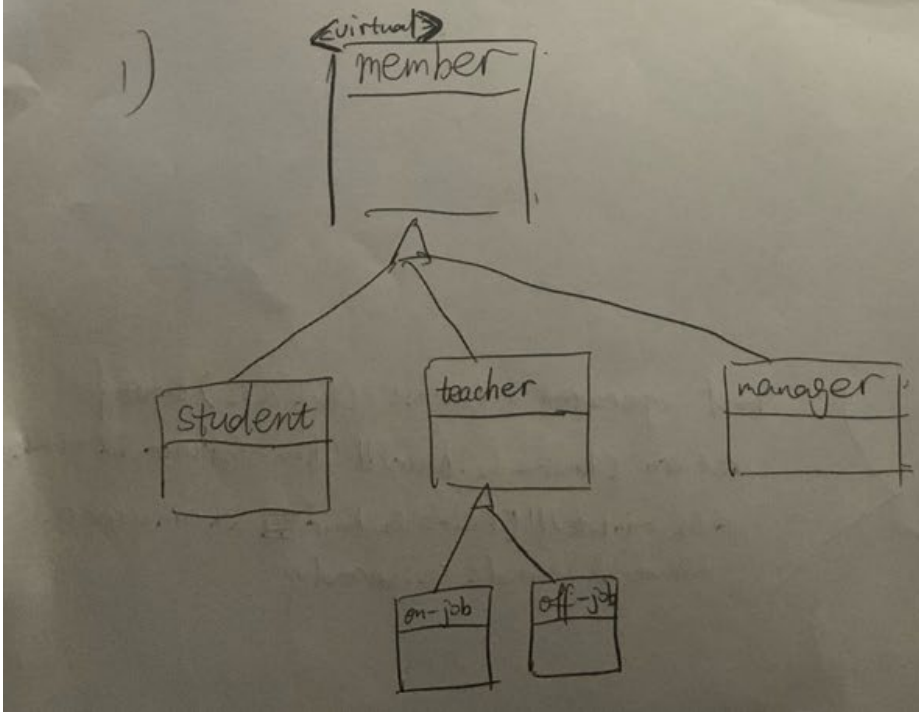
第3问，信息登记设置为成员函数，统计设置为非成员函数（但设为友元函数），是一种便于实现的方法，每次对给定群体调用这个非成员函数即可得到结果

第1问

学生一个类吧，另外两个老师和管理人员就按照那个图来。具体的研究生和本科生就在构造对象时分开吧

可以把硕士归为一个类。

这样的优点：结构清晰，便于管理，而且长远考虑，具有更强的可扩展性，比如本科生类可能以后会增加字段，函数等等。缺点：代码更繁杂，对于当前需求来说是溢出的。



```
class Member    //定义基类
{
protected:
        string number, name, phoneNumber, IDNumber;
        bool gender;
        int idType;
        string country, province, city, community, detailedAddr;
        bool isAddrChanged;
        int time, temperature;
        vector<int> state;
public:
        Member(){}      ~Member(){};
        void setInfo(string num, string name, string pNum, string IDNum, bool gender, int idType);
        void setAddr(string country, string province, string city, string community, string detailedAddr);
        void setTemperature(int time, int temperature, int stateNum, int i);
        friend void totalHealthState(Member p[], int size, int i);
        friend int addrChangedNumber(Member p[]);
};
class Student :public Member{};
void totalHealthState(Member p[], int size, int i);
int addrChangedNumber(Member p[]);
```

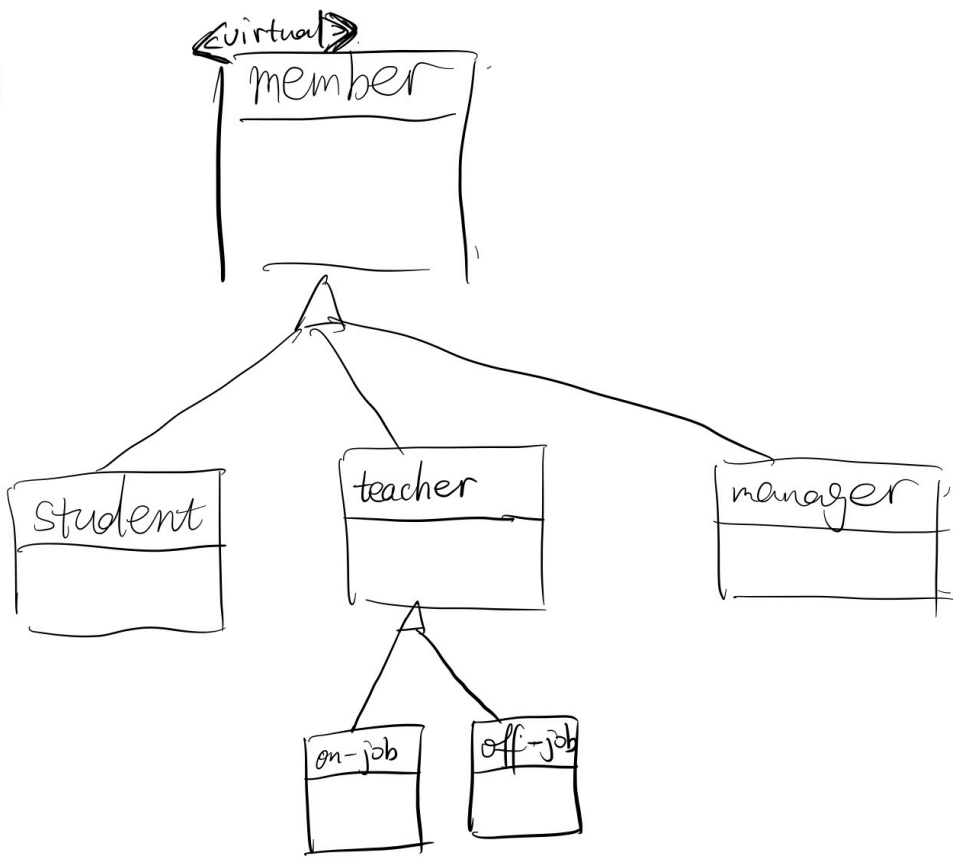
第3问，信息登记设置为成员函数，统计设置为非成员函数（但设为友元函数），是一种便于实现的方法，每次对给定群体调用这个非成员函数即可得到结果

第1问

学生一个类吧，另外两个老师和管理人员就按照那个图来。具体的研究生和本科生就在构造对象时分开吧

可以把硕士归为一个类。

这样的优点：结构清晰，便于管理，而且长远考虑，具有更强的可扩展性，比如本科生类可能以后会增加字段，函数等等。缺点：代码更繁杂，对于当前需求来说是溢出的。



考试注意事项:

递归，运算符重载，指针必考。

复数类重载

宏 define A B+C

D= A*6

B=2 C=3

易错 B+C*6 = 20

标识符规则.

位运算

do while 至少执行一次

语法问题多.

求阶乘 符号 int
阶乘结果 int } × 会溢出.

long double ✓

i!=1o1 !=号. 有序类型.

四. 程序编程 → ① 函数.(递归函数编程) ② 类及相关函数(运算符重载)

五. 分析界面.

不用实验代码.

答题要求似大作业报告，类结构，设计类和建立独立函数 及类之间关系

可能: 动态数组: 指针模拟数组.

考试中去.

问答. 4×4 概念.

表达式结果 ++ --

指针运算

改错 10 (5×2) 循环控制变量. 不能用浮点型. unsigned. 全局变量.

除数为0. 运行时会报警告.

数组下标. 溢出越界

const参加了函数重载

(仅作语法错)

程序阅读. 构造函数执行顺序

数组指针地址 形参

结果对. 分全拿

分析功能: 递归: 结果&功能

位运算: 无符号补0

有→补符号.

go to 不规范.