

北京师范大学2019~2020学年第一学期期末考试试卷 (A卷)

课程名称：C++程序设计 任课教师姓名：申佳丽余乐军

卷面总分：100分 考试时长：100分钟 考试类别：闭卷☒ 开卷☐ 其他☐

院 (系): _____ 专 业: _____

年级： — —

姓名：_____ 学号：_____

题号	第一题	第二题	第三题	第四题	总分
得分					

阅卷教师（签字）：_____

考试要求：

- 严格遵守考试纪律，考试过程中不得使用网络、优盘和手机等电子设备。
- 将每道题的解答写入答题区。
- 先浏览全部试题，按照适合自己的顺序答题。请勿在某一道题上花费过多时间。
- 写出答题过程，按步骤给分。
- 解答完毕后，请转为pdf文件提交。

一. 单项选择题 (每小题 2分, 共 20分)

1. class Test的默认析构函数是 _____ .
- A. void ~Test() {} B. Test() {}
C. Test ~Test() {} D. ~Test() {}

2. int a(10); 以下_____定义了 a的引用。
- A. int* p=&a; B. *a
C. int& r=a; D. &a

3. double * a=new double[10]; 以下_____正确收回了数组 a的空间。
- A. delete a; B. delete a[10];
C. delete[] a; D. delete a[];

4. class B: public A, A类的 private数据在 B类中是_____ 的。
- A. private B. 不可访问
C. protected D. Public

5. 程序部分代码如下

```
class A
{
public:
    void Output()
    {
        cout << "A ";
    }
};
class B : public A
{
public:
    void Output()
    {
        cout << "B ";
    }
};
```

按以上类 A和类 B的定义, 则以下代码的输出是_____ 。

```
B b;
A& a = b;
A* p = &b;
a.Output();
p->Output();
```

- A. A A B. B A
C. B B D. A B

二、多项选择题（每小题 2分，共 20分）.

- 以下 _____ 调用 class Person 的拷贝构造函数。
A. Person a[5];
B. Person a;
C. Person a; Person b(a);
D. Person a; Person b=a;
- 以下创建 class Sample 对象的语句, _____ 调用 Sample 的无参构造函数。
A. Sample one;
B. Sample one(1);
C. Sample arr[5];
D. Sample* ptr=new Sample;
- 体现“模块独立性更高”的技术有 _____。
A. 使用全局变量
B. 在类设计中, 把成员数据和成员函数封装在一起
C. 使用 const
D. 用 private、protected、public 限制外界对类成员的访问权限
- 实现代码重用性的技术有 _____。
A. STL
B. const
C. 模板
D. 继承
- 以下函数设计了形参默认值, 其中正确的是 _____。
A. void Set(int x=0, int y=0,int z=0);
B. void Set(int x=0, int y,int z=0);
C. void Set(int x, int y=0,int z=0);
D. void Set(int x=0, int y=0,int z);
- 为了防止被重复包含, 类声明文件 (.h 文件) 需要用到的条件编译命令有
A. _____。
#endif
B. #if
C. #ifndef
D. #define
- 以下 _____ 是可以重载的运算符是
A. > << >>
B. + - .* --
C. ?: % / sizeof
D. + <= [] ++
- swap 函数的功能是交换 main 函数中的两个整型变量, 以下 _____ 是正确的。
A. void swap(int& x, int& y);

- B. void swap(int* px, int* py);
- C. void swap(int x, int y);
- D. int swap(int x, int y);

9. cout<<setw(10)<<"bnu"<<endl; 需要_____。

- A. #include <vector>
- B. #include <algorithm>
- C. #include <iostream>
- D. #include <iomanip>

10.类的定义如下,

```
class Base
{
public:
    float GetDate() = 0;
};
```

函数“float GetDate()”是 _____, class Base是 _____。

- A. 抽象类
- B. 静态函数
- C. 虚基类
- D. 纯虚函数

三 . 程序综合 (共 30分)

1. 程序填空友元函数 (4分)

```
#include <iostream>
#include <cmath>
using namespace std;
class Point
{ double m_x, m_y;
public:
    Point(double x = 0, double y = 0) :m_x(x), m_y(y) {}
    (1)
};
double distance(const Point& p1, const Point& p2)//计算两点间距离
{ double x = p1.m_x - p2.m_x;
  double y = p1.m_y - p2.m_y;
  (2)
}
int main()
{ Point a(0, 0), b(1,1);
  cout << distance(a, b) << endl; //输出1.41421
  return 0;
}
```


2. 写出程序的输出 STL (3分)

```
#include<iostream>
#include<list>
using namespace std;
int main()
{   int i;
    list<int> L2;
    list<int>::iterator iter;
    int a2[] = { 30,40,50,60,60,60,80 };
    for (i = 0; i < 7; i++)
        L2.push_front(a2[i]);
    int sum = 0;
    for (iter = --L2.end(); iter != L2.begin(); iter--)
    {   cout << *iter << " ";
        sum += *iter;
    }
    cout << "\nL2: sum=" << sum << endl;
    int data = 0;
    return 0;
}
```

-(3)-

3. 程序填空 static (6分)

```
#include <iostream>
using namespace std;
class Circle
{private:
    int m_x, m_y, m_r;
    static int m_count;           //点的个数
public:
    Circle(int x = 0, int y = 0, int r = 0) :m_x(x), m_y(y), m_r(r)
    {   -(4)-
    }
    ~Circle()
    {   m_count--;
    }
    static int GetCount()
    {   return m_count;
    }
};
-(5)-
int main()
{   Circle c1, c2(3, 4, 5);
    cout << -(6)- << endl; //输出2
    return 0;
}
```

4.程序填空函数模板（4分）

```
#include <iostream>
#include <string>
using namespace std;
(7)
_____
_____
_____
_____

int main()
{   int A[10] = { 1,2,3,98,4,5,6,7,8,9 };
    double B[5] = { 1.2,-3.2,8.9,0,2 };
    string C[6] = { "abc","ok","cplus","java","school","123" };
    cout<<Max(A,10)<<" "<<Max(B,5)<<" "<<Max(C,6)<<endl; //输出98 8.9 school
}
```

5.程序填空继承（4分）


```

#include <iostream>
using namespace std;
class Person
{private:
    string m_name;
    int m_age;
public:
    void Set(string name, int age)
    { m_name = name;
      m_age = age;
    }
    void Show()
    { cout << m_name << " " << m_age << " ";
    }
};
class Student : public Person
{private:
    double m_score;
public:
    void Set(string name, int age, double score)
    { (8)
      m_score = score;
    }
    void Show()
    { (9)
      cout << m_score;
    }
};
int main()
{ Student s;
  s.Set("LiXiao", 20, 90);
  s.Show(); //输出 : LiXiao 20 90
  cout << endl;
  return 0;
}

```

带格式的:下划线,字体颜色:自动设置

6. 程序填空 const (3分)

```

#include <iostream>
using namespace std;
class MyStore
{private:
    int m_array[10];
public:
    MyStore(int* a)
    { for (int i = 0; i < 10; i++)
        m_array[i] = a[i];
    }
    (10)
    _____
    _____
    _____
};
int main()
{ int arr[10] = { 1,2,3,4,5,6,7,8,9,10 };
  const MyStore s(arr);
  s.Display();           //输出 1 2 3 4 5 6 7 8 9 10
  return 0;
}

```

7.程序改错（6分）

在错误语句的右边以“//error”的方式标注，简要说明原因；完善类设计以解决错误。请不要改变已有的类设计。

```

#include <iostream>
using namespace std;
class Teacher
{private:
    string m_name;
public:
    string GetName()
    { return m_name;
    }
};
int main()
{ Teacher xu;
  xu.m_name = "XuXiaoHong";
  cout << xu.GetName() << endl;   //期望输出: XuXiaoHong
  return 0;
}

```

(11)

四. 程序设计 (每小题 15分, 共 30分)

1. 学生参加考试, 所有的学生必考语文、数学、英语, 选考文科的同学加试历史、地理、政治, 选考理科的同学加试物理、化学、生物。

基类“基础科目”BaseSubject, 包含数据成员: 学生姓名、语文、数学、英语成绩, 包含成员函数: 构造函数、获得姓名函数、析构函数、平均分函数、总分函数。其中析构、平均分、总分是虚函数。

“基础科目”公有派生子类“文科科目”(SocialScience), 增加数据成员: 历史、地理、政治成绩, 包含成员函数: 构造函数、析构函数、平均分函数、总分函数。其中析构、平均分、总分是虚函数。

“基础科目”公有派生子类“理科科目”(Science), 增加数据成员: 物理成绩、化学成绩、生物成绩, 包含成员函数: 构造函数、析构函数、平均分函数、总分函数。其中析构、平均分、总分是虚函数。

已经给出了“基础科目”BaseSubject、main()函数, 以及程序的输出结果。

请补充公有派生子类“理科科目”(Science)的设计, 以及函数 void Show(BaseSubject& stu)的实现。

```
#include <iostream>
using namespace std;
class BaseSubject
{
private:
    string m_strName;
    float m_ch;      //语文
    float m_ma;      //数学
    float m_en;      //英语
public:
    BaseSubject(string name = "XXX", float ch = 0, float ma = 0, float en = 0);
    string GetName();
    virtual ~BaseSubject();
    virtual float Total();
    virtual float Average();
};

BaseSubject::BaseSubject(string name, float ch, float ma, float en) :
m_strName(name), m_ch(ch), m_ma(ma), m_en(en)
{
}

string BaseSubject::GetName()
{
    return m_strName;
}

BaseSubject::~~BaseSubject()
{
}

float BaseSubject::Total()
{
    return m_ch + m_ma + m_en;
}

float BaseSubject::Average()
{
    return (m_ch + m_ma + m_en) / 3;
}
```

```
void Show(BaseSubject& stu);
int main()
{   SocialScience wanghong("WangHong", 85, 90, 93, 88, 95, 76);
    Science liuli("Liuli", 88, 95, 90, 91, 92, 87);
    Show(wanghong);
    Show(liuli);
    return 0;
}
```

程序输出：

```
WangHong 527 87.8333
Liuli 543 90.5
```

2. 简单日期类 YearMonth,只描述年、月值。包含数据成员：年、月；成员函数：构造函数、格式化函数、重载输出运算符<<, 输入运算符>>, 加法+ (成员函数)、减法- (成员函数、被减数大于等于减数)、比较<= (成员函数)。

已经给出了类声明、部分函数实现、main()和两次运行结果。

请补充以下函数的实现：重载输出运算符 <<, 输入运算符>>, 加法+ (成员函数)、减法- (成员函数、被减数大于等于减数)、比较<= (成员函数)。

```
#include <iostream>
using namespace std;
class YearMonth
{private:
    unsigned m_year, m_month;
public:
    YearMonth(unsigned y = 1, unsigned m = 1);
    void Format();
    YearMonth operator+(YearMonth ym);
    YearMonth operator-(YearMonth ym);
    bool operator<=(YearMonth ym);
    friend ostream& operator<<(ostream& out, YearMonth ym);
    friend istream& operator>>(istream& in, YearMonth& ym);
};
YearMonth::YearMonth(unsigned y, unsigned m) :m_year(y), m_month(m)
{
    Format();
}
void YearMonth::Format()
{
    if (m_month > 12)
    {
        m_year += m_month / 12;
        m_month = (m_month - 1) % 12 + 1;
    }
}
```

```
int main()
{
    YearMonth date1, date2, date3;
    cout << "Input 2 dates(year-month), year>0, month>0: ";
    cin >> date1 >> date2;
    date3 = date1 + date2;
    cout << date1 << " + " << date2 << " = " << date3 << endl;
    date3 = date1 - date2;
    cout << date1 << " - " << date2 << " = " << date3 << endl;
    if (date1 <= date2)
        cout << date1 << " <= " << date2 << endl;
    else
        cout << date1 << " > " << date2 << endl;
    return 0;
}
```

程序的 2次运行：(下划线表示输入)

```
Input 2 dates(year-month), year>0, month>0: 2019-12 2015-3
2019-12 + 2015-3 = 4035-3
2019-12 - 2015-3 = 4-9
2019-12 > 2015-3
```

Input 2 dates(year-month), year>0, month>0: 2015-3 2019-12

2015-3 + 2019-12 = 4035-3

the First YearMonth Must >= the Second YearMonth