**2017—2018学年第二学期《面向对象程序设计》期末考试试题答案**

一、单项选择题（每题2分，共20分）

1-5 CDACB 6-10 BACBB

二、程序分析题

1.读程序，写运行结果。（10分，每行5分）

0A普通构造

0A析构

2.读程序，写结果。（10分，每行2分）

① 第3行using namespace std 改为 using namespace std;

② 第7行private:改为 public:

③ 第8行 Student(string name = "未知") mName (name) 改为 Student(string name = "未知") :mName(name)

④ 第20行 加上 int Student::count = 0;

⑤ 第26行 加上 delete arr;

3.读程序，写运行结果。（10分，每行2分）

① int &a,int &b

② int temp = \*a;

③ \*b = temp;

④ max = arr[0];

⑤ int i = 1; i < n; i++ 或者 int i = 0; i < n; i++

三、编程题。

1. 写一个point类来存储空间中的点。（10分）

#include <iostream>

using namespace std;

class Point

{

public:

void set(double aX, double aY)

{

m\_dX = aX;

m\_dY = aY;

}

double getX()

{

return m\_dX;

}

double getY()

{

return m\_dY;

}

private:

double m\_dX;

double m\_dY;

};

2.实现myAdd函数的重载。（10）

#include <iostream>

#include <string>

using namespace std;

int myAdd(int a,int b)

{

return a+b;

}

double myAdd(double a, double b)

{

return a + b;

}

string myAdd(string a, string b)

{

return a + b;

}

3. 请利用C++继承实现代码复用。（15）

#include <iostream>

using namespace std;

class Vehicle

{

public:

Vehicle( double aVWe, int aVWh);

protected:

double m\_dWeight;

int m\_iWheels;

};

Vehicle::Vehicle( double aVWe, int aVWh)

: m\_dWeight(aVWe), m\_iWheels(aVWh)

{

}

class Car :public Vehicle

{

public:

Car(double aVWe, int aVWh, int aCP);

void PrintData();

private:

int m\_iPassengers;

};

Car::Car( double aVWe, int aVWh, int aCP)

:Vehicle( aVWe, aVWh), m\_iPassengers(aCP)

{

}

void Car::PrintData()

{

cout << "车 重：" << m\_dWeight << "吨" << endl

<< "车轮数：" << m\_iWheels << " 个" << endl

<< "载客量：" << m\_iPassengers << " 人" << endl;

}

int main(void)

{

Car c1( 1.5, 4, 6);

c1.PrintData();

return 0;

}

4. 编写字符串CString类（15分）

#include <iostream>

using namespace std;

class CString

{

public:

CString(const char \* const p = "ABC"){

m\_p = new char[strlen(p) + 1];

strcpy(m\_p, p);

}

~CString()

{

delete[]m\_p;

}

CString(const CString &r){

m\_p = new char[strlen(r.m\_p) + 1];

strcpy(m\_p, r.m\_p);

}

CString & operator=(const CString &r)

{

delete[]m\_p;

m\_p = new char[strlen(r.m\_p) + 1];

strcpy(m\_p, r.m\_p);

return \*this;

}

void display()

{

cout << m\_p << endl;

}

private:

char \* m\_p;

};

int main(void)

{

CString s1("Hello");

CString s2(s1);

s2.display();

CString s3;

s3.display();

return 0;

}