**一、单项选择题**(每小题2分，共20分)

1.C 2.D 3.A 4.C 5.D

6.C 7.B 8.D 9.B 10.B

**二、程序分析题**（每小题10分，共30分）

**1.10分**

CON NORM

CON with a num

Display

Display

DES

DES

**2.10分（找对一个得1分，改对一个得1分）**

1. A( int ax) :x(ax){} 改为 A( int ax = 0) :x(ax){} （或者A st; 改为 A st（1）;）

2. this.x = aRef.x; 改为 this->x = aRef.x;

3. } 改为 };

4. 全局域中添加 int A ::KST;

5. A p = new A; 改为 A \*p = new A;

6. A \*p = new A； 语句后面添加 delete p；

**3.10分（每空2分）**

① int &a,int &b

② int temp = \*a;

③ \*b = temp;

④ max = arr[0];

⑤ int i = 0; i < n;i++

**三、编程题**（共50分）

**1.10分（答案不唯一，仅供参考）**

#include <iostream>

#include <cmath>

using namespace std;

class Point

{

public:

Point(int aX = 0,int aY = 0,int aZ = 0);

~Point();

double EuclidDistance(Point p1,Point p2);

int m\_iX;

int m\_iY;

int m\_iZ;

};

Point::Point(int aX,int aY,int aZ)

{

m\_iX = aX;

m\_iY = aY;

m\_iZ = aZ;

}

Point::~Point()

{

}

double Point::EuclidDistance(Point p1,Point p2)

{

int dX = p1.m\_iX - p2.m\_iX;

int dY = p1.m\_iY - p2.m\_iY;

int dZ = p1.m\_iZ - p2.m\_iZ;

return sqrt(dX\*dX + dY\*dY + dZ\*dZ);

}

int main(void)

{

Point p1,p2(3,4,5);

cout << p1.EuclidDistance(p1,p2) << endl;

return 0;

}

**2. 10分（答案不唯一，仅供参考）**

#include <iostream>

#include <string>

using namespace std;

class Student

{

public:

Student(string aNum = "",string aName = "",string aSex = "");

private:

string m\_strNum;

string m\_strName;

string m\_strSex;

};

Student::Student(string aNum,string aName,string aSex)

{

m\_strNum = aNum;

m\_strName = aName;

m\_strSex = aSex;

}

Student stu1("01","张三","男");//全局区

int main(void)

{

Student \*pstu = new Student("02","李四","女");//堆区

Student stu3("03","王五","男"); //栈区

delete pstu;//释放堆区内存

return 0;

}

**3.10分（本答案仅供参考，请酌情给分）**

#include <iostream>

using namespace std;

class Mouse

{

private:

Mouse() {}

static Mouse \*m\_pInstance;

public:

static Mouse \*GetInstance()

{

if(m\_pInstance == NULL)

m\_pInstance = new Mouse;

return m\_pInstance;

}

};

Mouse \*Mouse::m\_pInstance = NULL;

int main(void)

{

Mouse\* p1 = Mouse::GetInstance();

Mouse\* p2 = p1->GetInstance();

return 0;

}

**4. 20分（答案不唯一，仅供参考）**

#include <iostream>

using namespace std;

#define ElemType int

class Node

{

public:

Node();

Node \*next;

ElemType data;

};

Node::Node()

{

next = NULL;

data = 0;

}

class Stack

{

public:

Stack();

~Stack();

void Push(ElemType e);

bool Pop(int &e);

private:

Node \*base;

Node \*top;

int m\_iLen;

};

Stack::Stack()

{

base = new Node;

top = base;

m\_iLen = 0;

}

Stack::~Stack()

{

Node \*p = base;

while (p)

{

Node \*q = p->next;

delete p;

p = q;

}

}

void Stack::Push(ElemType r)

{

Node \*p = new Node;

p->data = r;

top->next = p;

top = p;

m\_iLen++;

}

bool Stack::Pop(int &e)

{

if(base == top) return false;

Node \*p = base;

for (int i = 0; i < m\_iLen-1; i++)

{

p = p->next;

}

top = p;

p = p->next;

e = p->data;

delete p; //释放被删除的结点

m\_iLen--; //长度减一

top->next = NULL;

return true;

}

int main(void)

{

Stack st1;

for(int i = 1;i < 10; i++) st1.Push(i);

int e = -1;

for (int i = 1; i < 16; i++)

{

st1.Pop(e);

cout << e << endl;

e = -100;

}

return 0;

}