2017-2018学年第一学期《面向对象程序设计C++》课试题A卷

一、单项选择题（每题2分，共20分）

1-5 CCBDD 6-10 CCCDC

二、程序分析题

1.读程序，写运行结果。（10分，每行2分）

abc

0

0

1

1

2.读程序，写结果。（10分，每行1分）

320

100

1000

100

100

1000

des

des

des

des

3.读程序，写运行结果。（10分，每行2分）

double

Base::doPrint

int

int

Derived:: doPrint

三、编程题。

1.按照下列类声明，实现String类。

#include <iostream>

using namespace std;

class String

{

public:

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

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

strcpy(m\_p, p);

}

~String()

{

delete[]m\_p;

}

String(const String &r){

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

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

}

String & operator=(const String &r)

{

delete[]m\_p;

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

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

return \*this;

}

bool operator==(const String &r)const

{

return strcmp(m\_p, r.m\_p) ? false : true;

}

const String operator+(const String & r)const

{

String temp;

delete[]temp.m\_p;

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

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

strcat(temp.m\_p, r.m\_p);

return temp;

}

void display()

{

cout << m\_p << endl;

}

private:

char \* m\_p;

};

2.认真阅读下面要求，完成本次任务代码的编写。（20）

#include <iostream>

using namespace std;

class Node

{

public:

int data;

Node \*next;

};

class List

{

public:

List()

{

head = NULL;

count = 0;

}

List(const List & list);

void insert(int value);

int &operator[](const int i);

List &operator=(const List & list);

~List();

private:

Node \*head;

int count;

};

List::~List()

{

Node \*p = head;

for (int i = 0; i < count + 1; i++)

{

head = head->next;

delete p;

p = head;

}

}

void List::insert(int value)

{

Node \*newNode = new Node;

newNode->data = value;

newNode->next = NULL;

Node \*p = head;

if (!head)

{

head = newNode;

return;

}

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

{

p = p->next;

}

p->next = newNode;

count++;

}

List::List(const List & list)

{

head = new Node;

count = 0;

Node \*p = list.head;

for (int i = 0; i < list.count; i++)

{

p = p->next;

insert(p->data);

}

}

List &List::operator=(const List & list)

{

Node \*p = head;

p = p->next;

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

{

Node \*q = p->next;

delete p;

p = q;

}

count = 0;

head->next = NULL;

Node \*q = list.head;

for (int i = 0; i < list.count; i++)

{

q = q->next;

insert(q->data);

}

return \*this;

}

int & List::operator[](const int index)

{

Node \*p = head;

if (index < 1 || index > count)

{

cout << "下标越界" << endl;

exit(1);

}

for (int t = 0; t < index; t++)

{

p = p->next;

}

return p->data;

}

3.在不修改Test类的情况下，通过编写程序直接或间接调用Test类的所有成员函数。（10）

#include <iostream>

using namespace std;

class Test

{

public:

void Display() const

{

}

protected:

Test(int aVal){

val = aVal;

}

void Function()

{

val = 0;

}

private:

int val;

};

class MyClass:public Test

{

public:

MyClass(int a):Test(a){}

void display() const{

Display();

}

void function()

{

Function();

}

};