C++面向对象程序设计（甲）（ B ）卷

注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交

一、判断题（对的打“√”，错的打“×” ）（每题 1 分，共计 10 分）

题　号	1	2	3	4	5
答　案					
题　号	6	7	8	9	10
答　案					

二、单项选择题（每题 2 分，总计 20 分）

题　号	1	2	3	4	5
答　案					
题　号	6	7	8	9	10
答　案					

三、程序填空题（每空 2 分，共计 12 分）

①_____

②_____

③_____

④_____

⑤_____

⑥_____

四、阅读程序题（共计 30 分）

1. 阅读该程序，给出程序的输出结果。（6 分）

2. 阅读该程序，给出程序的输出结果。（6 分）

3. 阅读该程序，给出程序的输出结果。（6 分）

4. 阅读该程序，给出程序的输出结果。（6 分）

5. 阅读该程序，给出程序的输出结果。（6 分）

五、编程题（共计 28 分）

1. 按照要求，编写程序（12 分）

2. 按照要求，编写程序（16 分）

C++面向对象程序设计（甲）（ B ）卷

注意：答案直接写在答题纸上，答在试卷上无效，考试后答题纸和试卷一同上交

一、判断题（对的打“√”，错的打“×” ）（每题 1 分，共计 10 分）

1. cout 的默认输出对象是键盘，cin 的默认输入对象是屏幕。
2. 使用 new 运算符创建数组时，可以为该数组指定初始值。
3. 内联函数的定义必须出现在第一次调用内联函数之前。
4. 在用 class 定义一个类时，数据成员和成员函数默认的访问权限是 public。
5. 构造函数可以设置默认参数。
6. 类的析构函数的作用是对象的初始化。
7. 只有常成员函数才可以操作常对象。
8. 模板类与类模板的意义完全相同。
9. 若类 Y 是类 X 的私有派生类，类 Z 是类 Y 的公有派生类，则类 Z 不能访问类 X 的公有成员和保护成员。
10. C++中设置虚基类的目的是实现运行时的多态。

二、单项选择题（每题 2 分，总计 20 分）

1. 下列关于面向对象概念的描述中，错误的是（ ）。
   A. 面向对象方法比面向过程方法更加先进
   B. 面向对象方法中使用了一些面向过程方法中没有的概念
   C. 面向对象方法替代了结构化程序设计方法
   D. 面向对象程序设计方法要使用面向对象的程序设计语言
2. 在函数的引用调用中，函数的实参和形参分别应是（ ）。
   A. 变量值和变量　　B. 地址值和指针
   C. 变量名和引用　　D. 地址值和引用
3. void Set(A &a); 是类 A 中一个成员函数的说明，其中 A &a 的含义是（ ）。
   A. 类 A 的对象引用 a 作该函数的参数
   B. 类 A 的对象 a 的地址值作函数的参数
   C. 表达式变量 A 与变量 a 按位与作函数参数
   D. 指向类 A 对象指针 a 作函数参数
4. 下列关于常成员的描述中，错误的是（ ）。
   A. 常成员是用关键字 const 说明的
   B. 常成员有常数据成员和常成员函数两种
   C. 常数据成员的初始化是在类体内定义它时进行的
   D. 常数据成员的值是不可以改变的
5. 下列关于 this 的描述中，错误的是（ ）。
   A. this 是一个由系统自动生成的指针
   B. this 指针是指向对象的
   C. this 指针在用对象引用成员函数时系统创建的
   D. this 指针只能隐含使用，不能显式使用
6. 下列关于运算符重载的描述中，错误的是（ ）。
   A. 运算符重载不改变优先级
   B. 运算符重载后，原来运算符操作不可再用
   C. 运算符重载不改变结合性
   D. 运算符重载函数的参数个数与重载方式有关
7. 下列关于派生类的描述中，错误的是（ ）。
   A. 派生类至少有一个基类
   B. 一个派生类可以作另一个派生类的基类
   C. 派生类的构造函数中应包含直接基类的构造函数
   D. 派生类默认的继承方式是 public
8. 下列运算符中，不可以重载的是（ ）。
   A. &&　　B. &
   C. [ ]　　D. .*
9. 下列的成员函数中，纯虚函数是（ ）。
   A. virtual void f1() = 0　　B. void f1() = 0;
   C. virtual void f1() {}= 0　　D. virtual void f1() == 0;
10. 下列关于抽象类的描述中，错误的是（ ）。
    A. 抽象类中至少应该有一个纯虚函数　　B. 抽象类可以定义对象指针和对象引用
    C. 抽象类通常用作类族中最顶层的类　　D. 抽象类的派生类不再是抽象类

三、程序填空题（每空 2 分，共计 12 分）

1. 请在下面程序的横线处填上适当内容，以使程序完整,并使程序的输出为:

下列程序的运行结果如下:
Base's cons.
Derived's cons.
Derived's des.
Base's des.
根据结果将程序补充完整。

```
#include <iostream.h>
class Base
{
public:
```

```
    Base(){cout<<"Base's cons."<<endl;}
    ___(1)___{cout<<"Base's des."<<endl;}
};
class Derived:public Base
{
public:
    Derived(){cout<<"Derived's cons."<<endl;}
    ~Derived(){cout<<"Derived's des."<<endl;}
};
void main()
{
    Base *Ptr=___(2)___
    delete ptr;
}
```

2. 实现下列求字符串长度的函数。

```
int strlen(char *str)
{
    int len;
    len = 0;
    while (*str)
    {
        ______(3)______
        ______(4)______
    }
    return len;
}
```

3. 下面程序通过把类 Distance 声明为类 Point 的友元类来实现计算两点之间的距离。请在下面程序的横线处填上适当字句，以使程序完整.

```
#include<iostream>
#include<cmath>
using namespace std;
class Point
{
    double X,Y;
public:
    Point(double x,double y){X=x;Y=y;}
    ____________(5)____________;
};
class Distance
{
public:
    double Dis(Point& p1,Point& p2);
};
double Distance::Dis(Point& p1,Point& p2)
{
    double t;
    ____________________(6)____________________;
    return t;
}
void main(){
    Point p(10,10),q(20,20);
    Distance d;
    cout<<d.Dis(p,q)<<endl;
}
```

四、阅读程序题（共计 30 分）

1. 阅读该程序，给出程序的输出结果。(6 分)

```
#include<iostream.h>
class Test
{
    static int n;
public:
    Test(){n += 2;}
    ~Test(){n -= 3;}
    static int GetNum(){return n;}
};
int Test::n=1;
void main()
{
    Test t,*p;
    p = &t;
    cout<<"n="<<p->GetNum()<<endl;
    p = new Test;
    delete p;
    cout<<"n="<<Test::GetNum()<<endl;
}
```

2. 阅读该程序，给出程序的输出结果。(6 分)

```
#include<iostream.h>
class A
{
    int num;
public:
    A(){num=0;cout<<"A default constructor"<<endl;}
    A(int n){num=n;cout<<" A constructor,num=" <<num<<endl;}
    ~A(){cout<<"A destructor,num="<<num<<endl;}
};

void main()
{
    A a,*p;
    p=new A(2);
    a=*p;
    delete p;
    cout<<" Exiting main" <<endl;
}
```

3. 阅读该程序，给出程序的输出结果。(6 分)

```
#include <iostream.h>
class S
{
public:
    S()
    {
        PC=0;
    }
    S(S &s)
    {
        PC=s.PC;
        for(int i=0;i<PC;i++)
            elems[i]=s.elems[i];
    }
    void Empty()
    {
        PC=0;
    }
    int IsEmpty()
    {
        return PC==0;
    }
    int IsMemberOf(int n);
    int Add(int n);
    void Print();
private:
    int elems[100],PC;
};
int S::IsMemberOf(int n)
{
    for(int i=0;i<PC;i++)
        if(elems[i]==n)
            return 1;
    return 0;
}
int S::Add(int n)
{
    if(IsMemberOf(n))
        return 1;
    else if(PC==100)
        return 0;
    else
    {
        elems[PC++]=n;
        return 1;
    }
}
void S::Print()
{
    cout<<'{';
    for(int i=0;i<PC-1;i++)
        cout<<elems[i]<<',';
    if(PC>0)
        cout<<elems[PC-1];
    cout<<'}'<<endl;
}
```

```
void main()
{
    S a;
    cout<<a.IsEmpty()<<endl;
    a.Print();
    S b;
    for(int i=1;i<=5;i++)
        b.Add(i);
    b.Print();
    cout<<b.IsMemberOf(3)<<endl;
    cout<<b.IsEmpty()<<endl;
    for(i=6;i<=10;i++)
        b.Add(i);
    S c(b);
    c.Print();
}
```

4. 阅读该程序，给出程序的输出结果。(6 分)

```
#include<iostream.h>
class Format
{
public:
    virtual void header(){cout<<"This is a head"<<endl;}
    virtual void footer(){cout<<"This is a footer"<<endl;}
    virtual void body(){cout<<"This is a body"<<endl;}
    void display(){header();body();footer();}
};
class MyFormat:public Format
{
public:
    void header(){cout<<"This is my header"<<endl;}
    void footer(){cout<<"This is my footer"<<endl;}
};
void main()
{
    Format * p;
    p=new Format;
    p->display();
    delete p;
    p=new MyFormat;
    p->display();
    delete p;
}
```

5. 阅读该程序，给出程序的输出结果。(6 分)

```
#include <iostream>
#include <vector>
using namespace std;

template <typename T>
void DisplayVector(vector<T>& vecInput)
{
   for(vector<T>::iterator iElement = vecInput.begin()
       ; iElement != vecInput.end ()
       ; ++ iElement )
     cout << *iElement << ' ';

   cout << endl;
}

int main ()
{
   vector <int> vecIntegers;

   vecIntegers.push_back (50);
   vecIntegers.push_back (1);
   vecIntegers.insert(vecIntegers.begin(), 987);
   vecIntegers.insert(vecIntegers.begin(), 1001);

   cout << "Vector contains " << vecIntegers.size () << " elements: ";
   DisplayVector(vecIntegers);

   vecIntegers.pop_back ();

   cout << "After a call to pop_back()" << endl;
   cout << "Vector contains " << vecIntegers.size () << " elements: ";
   DisplayVector(vecIntegers);
```

```
        return 0;
    }
```

五、编程题（共计 28 分）

1. 按下列要求编程（12 分）

定义一个描述矩形的类 Rectangle，包括的数据成员有宽（width）和长（length），并实现如下功能函数；

（1）矩形对象初始化；

（2）计算矩形周长；

（3）计算矩形面积；

（4）改变矩形大小；

（5）重载插入运算符（<<），实现矩形的长、宽、周长和体积的输出。

2. 设计一个 double 类型的数组类模板（CDblArray），要求 CDblArray 可以进行如下操作：（16 分）

（1）数组对象中存储元素的个数可以在定义的时候任意设定；

（2）可以重新设置数组对象中存储元素的个数（Resize）；

（3）可以通过下标运算符返回数组元素，并通过异常机制，处理下标越界异常；

（4）可以利用已知数组对象对另一个数组对象赋值和初始化；

（5）可以返回当前数组对象中存储元素的个数（Size）。